



UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Toward a Collaborative Platform for Hybrid Designs Sharing a Common Cohort

Jo Inge Arnes

A dissertation for the degree of Philosophiae Doctor

July 2023



Abstract

This doctoral thesis binds together four included papers in a thematical whole and is simultaneously an independent work proposing a platform facilitating epidemiological research.

Population-based prospective cohort studies typically recruit a relatively large group of participants representative of a studied population and follow them over years or decades. This group of participants is called a *cohort*. As part of the study, the participants may be asked to answer extensive questionnaires, undergo medical examinations, donate blood samples, and participate in several rounds of follow-ups. The collected data can also include information from other sources, such as health registers. In prospective cohort studies, the participants initially do not have the investigated diagnoses, but statistically, a certain percentage will be diagnosed with a disease yearly. The studies enable the researchers to investigate how those who got a disease differ from those who did not. Often, many new studies can be nested within a cohort study. Data for a subgroup of the cohort is then selected and analyzed. A new study combined with an existing cohort is said to have a *hybrid design*.

When a research group uses the same cohort as a basis for multiple new studies, these studies often have similarities regarding the workflow for designing the study and analysis. The thesis shows the potential for developing a platform encouraging the reuse of work from previous studies and systematizing the study design workflows to enhance time efficiency and reduce the risk of errors.

However, the study data are subject to strict acts and regulations pertaining to privacy and research ethics. Therefore, the data must be stored and accessed within a secured IT environment where researchers log in to conduct analyses, with minimal possibilities to install analytics software not already provided by default. Further, transferring the data from the secured IT environment to a local computer or a public cloud is prohibited. Nevertheless, researchers can usually upload and run script files, e.g., written in R and run in R-studio. A consequence is that researchers—often having limited software engineering skills—may rely mainly on self-written code for their analyses, possibly un-

systematically developed with a high risk of errors and reinventing solutions solved in preceding studies within the group.

The thesis makes a case for a platform providing a collaboration *software as a service* (SaaS) addressing the challenges of the described research context and proposes its architecture and design. Its main characteristic, and contribution, is the separation of concerns between the SaaS, which operates independently of the data, and a secured IT environment where data can be accessed and analyzed. The platform lets the researchers define the data analysis for the study using the cloud-based software, which is then automatically transformed into an executable version represented as source code in a scripting language already supported by the secure environment where the data resides.

The author has not found systems solving the same problem similarly. However, the work is informed by cloud computing, workflow management systems, data analysis pipelines, low-code, no-code, and model-driven development.

Acknowledgements

I would like to express my gratitude to the individuals and groups who have contributed to this doctoral journey.

First, I thank my main supervisor, Professor Alexander Horsch, for the inspiring conversations and motivation. His guidance has been invaluable in navigating the challenges of this project. I cannot overstate how crucial he has been. Without him, I would not have completed the journey.

Great appreciation also goes to co-supervisor Professor Alexander Hapfelmeier for his expert feedback on statistical topics. His expertise as a medical statistician and biometrician has been fundamental to the progression of the research.

Many thanks also go to Associate Professor Tonje Braaten at the Systems Epidemiology group at the Department of Community Medicine. I am very grateful for the collaboration and hope for future opportunities.

Further, I want to mention Professor Lars Ailo Bongo for making the Ph.D. project possible and Professor Torkjel M. Sandanger and Professor Eiliv Lund for being of help in developing my knowledge in epidemiology early in the project.

Thanks to the members of Associate Professor Dilip K. Prasad's Bio-AI group, the HDL group, and the Systems Epidemiology group for your friendly spirit and knowledge sharing. Working with you has been enriching. I also want to thank the staff and students I have gotten to know at the Department of Computer Science. The department has been a welcoming and productive environment, and I value being a part of this community.

Last but not least, I thank my loving and patient wife, my bright and beautiful little daughter, and my son, who has grown up to be an independent young man. You mean the world to me.

Jo Inge Arnes—Tromsø, July 2023

Contents

| | |
|--|-------------|
| Abstract | i |
| Acknowledgements | iii |
| List of Figures | ix |
| List of Tables | xi |
| Glossary | xiii |
| List of Abbreviations | xvii |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Key scenario | 4 |
| 1.3 Need for systematization of the work process | 5 |
| 1.4 The need for multi-tenancy | 7 |
| 1.5 Computer system design challenges | 7 |
| 1.6 Thesis statement | 12 |
| 1.7 Reading guide | 13 |
| I Background | 15 |
| 2 Epidemiology | 17 |
| 3 Bias, Associations, and Causality | 19 |
| 4 Study Designs | 21 |
| 4.1 Case-Control Studies | 21 |
| 4.2 Longitudinal Studies and Prospective Cohorts | 22 |
| 4.3 Hybrid Designs | 22 |
| 4.4 Cross-Sectional Design | 23 |

| | |
|--|-----------|
| 5 DNA, Genomics, and Omics | 25 |
| II Methods | 27 |
| 6 Cloud Computing | 29 |
| 7 Workflow Management Systems | 33 |
| 8 Low-Code and No-Code Development | 35 |
| III Toward a Platform for Hybrid Designs and Analysis Pipelines | 39 |
| 9 A Workflow for Designing Hybrid Studies and Data Analysis Pipelines | 41 |
| 9.1 Designing the study and pipeline | 42 |
| 9.2 Applying for data | 43 |
| 9.3 Executing the analysis pipeline | 44 |
| 10 Architecture and Design | 45 |
| 11 Study Workflow Use Case | 49 |
| 11.1 Study Designer | 50 |
| 11.2 Pipeline Designer | 50 |
| 11.3 Data Application Manager | 51 |
| 12 Building Runnable Pipelines | 53 |
| IV Summary of Research | 55 |
| 13 Overview | 57 |
| 14 Paper I | 59 |
| 15 Paper II | 61 |
| 16 Paper III | 63 |
| 17 Paper IV | 65 |
| 18 Concluding Remarks | 67 |

| | |
|---|------------|
| 18.1 Limitations | 68 |
| V Included Papers | 69 |
| Paper I: The Beauty of Complex Designs | 71 |
| Paper II: Autostrata: Improved Automatic Stratification for Coarsened Exact Matching | 99 |
| Paper III: Greedy Knot Selection Algorithm for Restricted Cubic Spline Regression | 109 |
| Paper IV: Schema-Based Priming of Large Language Model for Data Object Validation Compliance | 129 |
| Bibliography | 167 |

List of Figures

| | | |
|------|---|----|
| 1.1 | The figure is a concept map illustrating the related topics in the thesis. It also maps the included works to topic areas, showing how each covers an area, illustrated by a gray background bounded by a darker line. The concept nodes have background color gradients indicating the three fields; computer science (pink), epidemiology (blue), and statistics (green). Where the fields blend, so do the respective background colors. | 2 |
| 9.1 | The figure illustrates the three basic steps of the overall workflow. | 42 |
| 9.2 | The figure illustrates an expanded version of the overall workflow. | 42 |
| 10.1 | The figure illustrates the system's three primary environments: The cloud, the researcher's local computer, and the secured IT environment with a VM having access to the provisioned data for the study. | 46 |
| 10.2 | Five different user roles in the presented architecture | 47 |
| 10.3 | The figure is a UML component diagram giving a high-level view of the components in the architecture relevant to the study design workflow as used by a researcher role. | 48 |

List of Tables

| | | |
|-----|---|----|
| 1.1 | Defining attributes of the data analysis context. | 3 |
| 7.1 | Cloud providers and WFMS service | 34 |

Glossary

Analysis Pipeline An analysis pipeline comprises a series of automated steps applied in conjunction with a data analysis process, where each step transforms input data, and the output of one step becomes the input of the next until finished.

Concept Map A concept map shows a network of nodes representing concepts connected with links (lines) representing associations. The links can be directional, illustrated by arrowheads at the end of the lines. The links can also have verbs or words expressing the nature of the association. An example and explanation is provided in Lanzing (1998, p. 5).

Data Analysis Data analysis includes tasks such as modeling, regression, correlation, classification, clustering, forecasting, and visualization. However, it excludes the data wrangling and other steps preparing data for the analysis and post-analysis interpretation, documentation, and evaluation. (Runkler 2020, p. 2)

Data Analytics Data analytics is the application of computer systems to analyze large data sets to support decisions. (Runkler 2020, p. 2)

Data Object In this thesis, a data object is a region of storage containing values for an entity or concept in the problem domain, such as the registered data on a particular employee, shipped item, shopping list, or—more relevant to this thesis—the definition of a particular analysis pipeline. The data object is an instance of a datatype or structure comprised of attributes (variables) holding values. However, it has no behavior, that is, no functions/methods.

Data Provisioning Data provisioning is the process of making data available in an orderly and secure manner.

Data Steward A data steward is responsible for reviewing data access requests and managing how the data are used, ensuring that the data are handled responsibly and in accordance with ethical and legal guidelines.

Data Wrangling Data wrangling is the preprocessing of raw data per further processing needs, e.g., cleaning, filtering, completing, correcting, standardizing, transforming, and structuring raw data.

Epidemiology A widely accepted definition is "the study of the distributions and determinants of health-related states and events in specified populations and the application of this study to control of health problems." (Gordis 2014, p. 2)

Health Outcome A health outcome may be a disease, condition, death, event or a change in health status or behavior. (Alexander et al. 2015c)

Hybrid Design "Hybrid designs combine the elements of at least two basic designs, or extend the strategy of one basic design through repetition. Two popular hybrid designs are the case-cohort study and the nested case-control study. Both these designs combine elements of a cohort and case-control study. Another more recently developed hybrid design is called the case-crossover design." (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 39)

Low-Code Low-code development enable rapid application development through techniques that reduce the amount of code written, often including model-driven engineering, domain-specific languages, and drag-and-drop mechanisms. (Pinho, Aguiar, and Amaral 2023)

Model-Driven Development Model-driven development (MDD) is based on defining models in software (semi)automatically transformable into executable models usually interpretable by a model-execution engine or represented as source code (Brambilla, Cabot, and Wimmer 2012, pp. 26–33).

Multi-Tenant Multi-tenant means that several tenants, or consumers, share the same resources while being isolated in separate environments and precluded from knowing about or accessing the other tenants' environments.

No-Code No-code development is closely related to low-code but aims to remove the need to write code completely, e.g., using visual design tools.

Observational Studies In observational studies, the researchers make observations about exposures or treatments and health outcomes under investigation while avoiding affecting any variables, measured and unmeasured, or altering the environment of the subjects in the study. This contrasts experimental studies, where researchers manipulate certain variables,

such as the treatment given, in a controlled environment.

Pipeline Definition A pipeline definition is a data structure specifying and configuring a set of analysis pipeline steps and associated tasks. The pipeline definitions in the proposed platform do not include runnable code or commands. Instead, the platform builds separate runnable pipelines from pipeline definitions. The term is equivalent to *process definition* in Hollingsworth (1995). From a model-driven development perspective, a pipeline definition is a model that can be transformed into an executable model.

Runnable Pipeline The proposed platform produces runnable pipelines from pipeline definitions, which can be executed in a secured IT environment where restricted data are accessible. From a model-driven development perspective, a runnable pipeline is an executable model.

Schema "A description of the structure of some data, including its fields and datatypes. Whether some data conforms to a schema can be checked at various points in the data's lifetime, and a schema can change over time." (Kleppman 2017, p. 557).

Scripting Language A scripting language does not require up-front source code compilation before being run. Instead, a script engine interprets the source code or compiles it on the fly at run time. The latter is often called *just-in-time* (JIT) compilation (Aycock 2003). Many programming languages originally designed for either compilation or scripting can now be used for both.

Secured IT Environment An IT environment that incorporates and adheres to Information Security Management System (ISMS) principles. In the context of the epidemiological research data discussed in the thesis, it is additionally an IT environment ensuring that the data are protected by security and privacy controls and enabling the research to comply with acts, regulations, and ethical guidelines.

Study Design In analytical epidemiology, researchers use standard *study designs* as a foundation for designing studies and avoiding typical systematic errors when assessing relationships between treatments or exposures and health outcomes. Each design has its well-known strengths and weaknesses. Deciding on an appropriate study design is essential because many serious problems and mistakes at the design stage of a study cannot easily be rectified in subsequent stages (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 21).

Workflow A workflow is a repeatable series of steps in a work process, organized in a specified order from start to end, possibly including conditional branches, where each step includes one or more activities to execute before proceeding to the next. The steps can involve a combination of human and machine-based activities.

List of Abbreviations

- Alibaba** Alibaba Cloud
- AWS** Amazon Web Services
- Azure** Microsoft Azure
- BPMS** Business Process Management System
- CEM** Coarsened Exact Matching
- CWL** Common Workflow Language
- DB** Database
- DRAGEN** Dynamic Read Analysis for GENomics
- ECJ** European Court of Justice
- EEA** European Economic Area
- EU** European Union
- FPGA** Field Programmable Gate Array
- GATK4** Genome Analysis Toolkit 4
- GCP** Google Cloud Platform
- GDPR** General Data Protection Regulation
- HPC** High-Performance Computing
- HTS** High-Throughput Sequencing

- IDE** Integrated Development Environment
- ISM** Department of Community Medicine
- LLM** Large Language Model
- NIST** National Institute of Standards and Technology
- NLUI** Natural-Language User Interface
- NOWAC** Norwegian Women and Cancer
- RCS** Restricted Cubic Spline
- REK** Regional Committees for Medical and Health Research Ethics
- saas** Software as a Service
- SPA** Single-Page Application
- UI** User Interface
- UML** Unified Modeling Language
- US** United States
- VM** Virtual Machine
- WDL** Workflow Description Language
- WFMS** Workflow Management System



Introduction

This doctoral thesis proposes a system to facilitate collaboration within research groups in epidemiology and help systematize the study design process. Further, the thesis provides the thematical whole at the intersection of computer science, statistics, and epidemiology in which the four included papers partake, each individually having novel contributions. Figure 1.1 is a concept map showing how the thesis and papers fit in the complete picture.

1.1 Motivation

Epidemiologists increasingly use population-based prospective cohorts as a basis for additional analytical observational studies, which has considerable benefits (Chapters 2–4). For example, case-control studies are prone to selection and recall bias, but nesting the studies in a prospective cohort can significantly reduce these biases. Prospective cohort studies can span over years or decades and include more than a hundred thousand participants. Data for each participant are collected systematically from surveys, registers, biological samples, and other sources. One way of producing invitation lists for population-based prospective cohort studies is by sampling from a national population registry. The studies recruit and follow participants who initially are without any investigated diagnosis, but statistically, some percent of the cohort will receive a diagnosis per year. For these diagnosed participants, follow-up may also continue afterward. The Norwegian Women and Cancer (NOWAC) study is

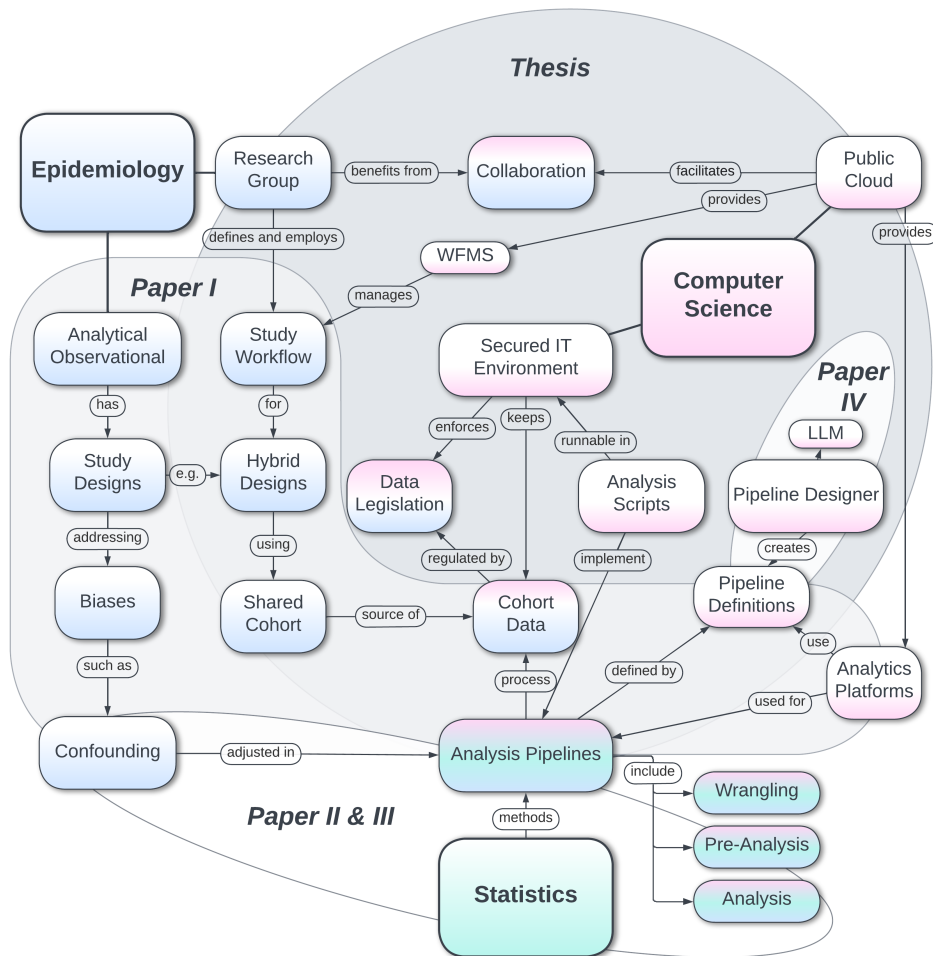


Figure 1.1: The figure is a concept map illustrating the related topics in the thesis. It also maps the included works to topic areas, showing how each covers an area, illustrated by a gray background bounded by a darker line. The concept nodes have background color gradients indicating the three fields; computer science (pink), epidemiology (blue), and statistics (green). Where the fields blend, so do the respective background colors.

one such population-based prospective cohort that includes 172,472 Norwegian women born between 1926–1965 (Gram et al. 2013; *Variabler* 2023). NOWAC is managed by the Systems Epidemiology group of the Department of Community Medicine (ISM) at UiT The Arctic University of Norway, which actively uses the cohort as a basis for hybrid designs in the research. Throughout the thesis, NOWAC is the primary example of a cohort suitable for hybrid designs. A background on epidemiology is found in Part I, whereas the first included paper, *The Beauty of Complex Designs*, provides a more thorough account of NOWAC and systems epidemiology.

It is not unusual for modern epidemiological studies to include data generated from biological material, and much focus is on high-throughput sequencing (HTS) data and *omics*, for example, genomics, epigenomics, and transcriptomics (Chapter 5). Consequently, hundreds of software packages and systems exist to explore and analyze these (Fjukstad 2019). However, this doctoral thesis focuses not on analysis pipelines for biomedical datasets. Instead, it focuses on workflows for designing studies and analysis pipelines using the same parent cohort as a basis and a platform providing a software as a service (saas) supporting these work processes. The platform's three central features are:

- Support for analysis pipelines adapted to the restrictions applying to epidemiological data
- Sharing of custom, reusable analysis pipeline tasks within a group
- User-friendly design workflows for novel studies and analysis pipelines

Although many systems and frameworks exist for workflow management and data analysis, a set of attributes particular to the discussed context poses challenges not addressed by existing systems. Table 1.1 lists the defining attributes.

Table 1.1: Defining attributes of the data analysis context.

| | |
|----|---|
| 1. | The data are sensitive, and acts, regulations, and ethical guidelines restrict the access and allowed storage environment. Researchers apply for access to specific subsets of the data. |
| 2. | The same cohort is a basis for numerous studies. |
| 3. | The studies use relatively small subsets of the cohort data. |
| 4. | The researchers usually process and analyze the data using a scripting language. There is potential for reusing script functions developed in preceding studies, but no systematic approach exists to sharing, managing, or applying the scripts. |
| 5. | Each study is unique but has commonalities with other studies in the same group. Despite differences, the studies are similar regarding the overall study design workflow. |
| 6. | Many of the researchers have limited experience with complex programming tasks. |

1.2 Key scenario

This section illustrates how the attributes listed in Table 1.1 are typically observed in practice. It is based on an actual study and is the key scenario for understanding the context and current situation the proposed system aims to improve.

Smoking exposure and differentially expressed genes in blood

A first-year Ph.D. student in systems epidemiology is researching differentially expressed genes associated with smoking in the blood of cancer-free women. The results can help control for confounder bias in subsequent research on lung cancer. A comprehensive smoking index (CSI) (Leffondré et al. 2006) must be computed per participant as part of the research. The participants are the controls from a nested case-control study. The data are a small subset of a large population-based prospective cohort with extensive questionnaires and blood samples provided by a data steward after the approval of a data access application. The data steward is also a member of the research group, but a formal application is still required.

The data for the subset are small enough to be loaded in working memory and analyzed in an instance of R-studio running in a virtual machine (VM) (Barham et al. 2003; Sugerman, Venkitachalam, and Lim 2001) in the secured IT environment by using a selection of custom scripts.

A complicating factor is that the cohort participants have answered several follow-up questionnaires in various versions, all including a range of smoking-related questions. In total, there are 77 different smoking-related variables. Thus, the dataset's many smoking-related variables require a complicated data wrangling script, which does not exist. A Ph.D. student from the Department of Computer Science is asked to participate in developing the smoking-variable data wrangling script. There are also other scripts unrelated to smoking needed in the study. For example, a version of an existing script written by the epidemiology Ph.D. student's supervisor in conjunction with another study is sent via email.

The described scenario is based on a real-world study, and the research was published in the paper *Gene expression in blood reflects smoking exposure among cancer-free women in the Norwegian Women and Cancer (NOWAC) postgenome cohort* (Baiju et al. 2021). The author of this thesis wrote the data wrangling scripts for the study and is credited as J.I. in the paper's Acknowledgements. A repository with the scripts and a 60 page report is available (<https://github.com/jo-inge-arnes/smoking-variables>).

1.3 Need for systematization of the work process

The scenario described in Section 1.2 is not unique. Although the study was successful, a more systematic approach can make the work process more efficient and lower the risks of unnecessary errors, discussed in this section.

Before being granted access to the data needed for a study, the researchers must apply for access to a specific subselection of variables and data. If the application is approved, a data steward provides the data. For NOWAC, the application instructions are found on the research group's official website (<https://uit.no/research/nowac/project?pid=797468>). It is mandatory to include a list of the variables needed for the study, a project description, and a publication plan. There are 12,105 available NOWAC variables (*Variabler 2023*) from questionnaires and registers, but access is granted solely to the variables listed in the application. Additionally, preserved blood and tissue samples exist for parts of the cohort. An opportunity for improvement is to develop a system facilitating and streamlining the application process for use within the group.

For processing data, the studies regularly need scripts tailored to the underlying cohort dataset, study designs, and analyses used by the group. Unfortunately, the development process can be time-consuming and difficult. For example, the data wrangling of smoking-variables described in Section 1.2 took weeks to develop. It required manually decoding the meaning of the different variables by comparing them to the fields in the original set of different questionnaires, available only on paper, and discussing how to combine the 77 variables correctly. Further, many participants answered inconsistently between follow-ups or changed their smoking status during the study. Because smoking is highly relevant to cancer studies, the same operation may be helpful in other studies within the research group. Thus, reusable scripts could save time and resources in research projects.

However, a reusable script is only helpful to the group if its existence is known to its members. Unfortunately, making reusable scripts known is often done in a word-of-mouth fashion. The danger in this unsystematic approach is that a script's existence may become known only to a few group members or forgotten. Thus, the group would benefit from a system acting as a hub for sharing and improving the discoverability of reusable scripts.

A system for sharing reusable scripts and other resources has several advantages besides discoverability. For example, a supervisor of the epidemiology doctoral student emailed a self-developed script. This approach is problematic because

changes to the script by the receiver, and further unsystematic distribution of these changes, will result in diverging branches of source code and likely defects not seen in the original or other branches of the script. Furthermore, testing, upgrading, and applying fixes to all script versions becomes difficult without a central overview of the existing versions. If someone makes corrections to the script without applying it to all other alternative versions, or the script is only partially reusable and requires customization on a source code level per study, the various script versions will have differences that make it challenging to maintain quality and correct defects across all versions. Thus, without a systematic and centralized approach to version control, it is likely that defects will get introduced or continue to exist uncorrected. A system acting as a hub for distributing reusable scripts can alleviate this problem. Additionally, the system could encourage documentation and testing of the scripts, which would further improve the quality.

By arranging reusable scripts in a logical sequence, an analysis pipeline can be constructed where each script represents a task and tasks follow each other in order. To effectively combine reusable scripts into an analysis pipeline, it is essential to establish data structures and interfaces for defining the pipeline, its steps, and its tasks. This ensures that the scripts are compatible and interact effectively. By creating a data structure representing a pipeline definition and framing the reusable scripts as tasks, it is possible to have a clear separation of concerns, ensuring that each script performs a distinct function within the well-defined steps of the analysis pipeline. While data analysis shares common elements across different fields, designing a study and an analysis pipeline in epidemiological research involve specialized steps. However, within the same research group, the typical study design workflow and the rough division of analysis-related tasks are relatively similar, making it possible to create a consistent workflow for designing analysis pipelines.

In the key scenario in Section 1.2, a research group member acting as data steward evaluated data access applications and was responsible for data provisioning. The group's existing manual workflow thus involves more than one human actor. The ability to distribute workflow steps between several human and automated actors with different roles is a common WFMS feature (Hollingsworth 1995), which can be employed to simplify the data application process. For example, a researcher can design a study in the first part of the workflow and then apply for data access. Next, the group member responsible for processing the application and providing the data could continue the workflow, possibly using applicant-specified scripts to select the requested data, before returning the workflow control to the researcher.

1.4 The need for multi-tenancy

The system described up to this point has concentrated on a single research group. However, other research groups likely have equivalent needs for defining common workflows, designing studies, sharing reusable scripts, and creating analysis pipeline. Rather than developing a system specifically for a single research group (tenant), which must allocate time and resources toward installing and managing an on-premise system, the system should instead be designed as a multi-tenant saas platform (Chapter 6). By designing the system as a cloud-based platform, many research groups can establish separate memberships, and the platform can offer the flexibility to tailor workflow templates and share reusable scripts per group. Employing such a saas service can reduce the management overhead, and the research groups can focus more on their core objectives.

1.5 Computer system design challenges

Section 1.3 described the problems of an unsystematic approach to managing scripts for studies and the benefits of introducing a system acting as a hub for reusable scripts. It also suggested that workflows and features informed by WFMS could improve the study design and management process. Further, the epidemiological studies discussed in this doctoral thesis are analytical observational studies; thus, data analysis is a central topic. A plethora of advanced frameworks, architectures, platforms, and cloud services exist for data analysis. However, the attributes listed in Table 1.1 give rise to challenges not fully covered by the existing solutions.

First and foremost, it is essential to be aware of the differences in how datasets must be handled in epidemiology and medical research compared to many other fields. These differences constitute **significant challenges** when designing the system. National and international acts, regulations, and research ethics restrict the collection, storage, access, transfer, and sharing of data containing sensitive personal and health-related information. Data analytics services in the public cloud naturally require data to be accessible in the cloud in one way or another. However, this is far from straightforward for the research data discussed here. In general, the General Data Protection Regulation (GDPR) (<https://gdpr.eu/>) imposes restrictions concerning the privacy and data protection for all individuals within the European Union (EU) and the European Economic Area (EEA), including Norway (*Regulations 2018*)—further, transferring personal data to countries outside the EU and the EEA is strictly regulated. The Safe Harbor Principles was a set of standards to govern personal data transfer between the EU and the United States (US), but the framework was invalidated

in 2015 by the European Court of Justice (ECJ) (*Federal Trade Commission Enforcement of the U.S.-EU and U.S.-Swiss Safe Harbor Frameworks* 2016). The EU-US Privacy Shield was established in 2016, but the ECJ also invalidated this agreement in 2020 (*Update on the Privacy Shield Framework* 2020). This affects cloud services that cannot guarantee where data are transferred through or stored. On a national level, in Norway, there are additional legislation pertaining to patient and health-related data, such as the Health Records Act (*Pasientjournalloven* 2022) and the Health Register Act (*Helseregisterloven* 2022). There are also the Norwegian Health Research Act (*The Health Research Act* 2020) and other regulations for health-related research. The Regional Committees for Medical and Health Research Ethics (REK) is the authority when applying for approval of research projects in Norway.

As an example, all of the following acts and regulations are relevant for NOWAC (*Variabler* 2023):

- The Public Health Act
- The Health Research Act
- Personal Health Data Filing System Act (The Health Register Act)
- The Personal Data Act
- The General Data Protection Regulation
- The Research Ethics Act
- The Patient Injury Act
- The Public Administration Act
- The Freedom of Information Act
- The Patient and Consumer Rights Act
- The Regulation Relating to Population-Based Health Surveys

The HUNT Cloud (<https://www.ntnu.edu/mh/huntcloud>) is an example of a cloud authorized for storing and accessing epidemiological research data, including data generated from human biological material, whereas data not from human biological material can alternatively be stored at TSD (<https://www.uio.no/english/services/it/research/sensitive-data/>), and some Microsoft Office 365 (<https://office.com>) subscriptions, but the range of available services

is limited and far from comparable to the common public clouds. These restrictions and limitations make it challenging to use cloud services in the context discussed in the thesis. The system must be designed to work with data stored in an appropriate, secured IT environment, such as the HUNT Cloud or TSD, without requiring data to be transferred to another cloud or on-premise system for processing.

The population-based cohort data used by the research groups, such as the NOWAC data, change relatively slowly and have a static structure. Also, the researchers commonly only use a smaller subset of the cohort data for each study. For example, in the context of the thesis, the vast majority of studies using HTS data are conducted in a setting of small data, including fewer than 100 participants (Holsbø 2019, p. 3). In contrast, modern data analytics platforms predominantly focus on big data and are architected to handle petabytes or more data. These big data platforms are for high-speed, real-time data processing and emphasize scalability, distributed processing, and fault tolerance. For example, Apache Kafka¹, Flink², Spark³, and Beam⁴ excel at voluminous, dynamic, and fast-paced flows of data. However, the complexities of these big data and streaming analytics platforms outweigh their advantages for small, slowly changing data. While big data analytics is transformative in the right context, it is not an optimal choice here.

In practice, the researchers mostly use a scripting language with good support for statistics, like R⁵, to work with the data. They are usually competent in statistics and know how to use the chosen scripting language for this purpose. Another reason is that data are accessed and analyzed by logging in to a secured IT environment where the possibilities for installing custom analytics platforms and software are limited or nonexistent, whereas using statistics software such as R-studio is usually unproblematic.

The researchers employ a combination of custom and reusable scripts to transform data in a series of steps, i.e., they create analysis pipelines. Further, each study is unique, but the overall workflows for the typical hybrid designs used within the research groups are relatively similar. There are numerous existing standards for defining workflows and analysis pipelines. Two popular languages for defining workflows, most notably used in bioinformatics, are Common Workflow Language (CWL)⁶ and Workflow Description Language

1. <https://kafka.apache.org/>
2. <https://flink.apache.org/>
3. <https://spark.apache.org/>
4. <https://beam.apache.org/>
5. <https://www.r-project.org/>
6. <https://www.commonwl.org/>

(WDL)⁷. WDL was developed by Broad Institute⁸ to define workflows utilizing the tools in their Genome Analysis Toolkit 4 (GATK4)⁹, which is a toolkit for variant discovery in high-throughput sequencing (HTS) data. Broad Institute has also developed Cromwell (Caetano-Anolles 2023b), an open-source workflow execution engine that supports both WDL and CWL. GATK4 includes several pre-defined *Best Practices* workflows (Caetano-Anolles 2023a), which can be run using the Cromwell + WDL pipelining solution. Best Practices workflows exist for whole genomes, exomes, gene panels, and RNAseq experimental designs. GATK4 is supported on a wide range of platforms and by major cloud providers, e.g., Amazon AWS, Microsoft Azure, Google Cloud, Alibaba Cloud, and IBM Cloud, and can be used in high-performance computing (HPC) environments¹⁰. For several tools in the toolkit, versions have been developed for Apache Spark. Further, DRAGEN-GATK¹¹ is a hardware-accelerated Field Programmable Gate Array (FPGA) (Koch, Ziener, and Hannig 2016, pp. 10–11) version based on Illumina’s Dynamic Read Analysis for GENomics (DRAGEN) Bio-IT platform¹². However, the workflows and analysis pipelines discussed in this thesis are on an altogether different abstraction level than the HTS workflows. The reason for describing GATK4, CWL, WDL, Best Practices workflows, and the Cromwell execution engine is to illustrate properties common to most of the popular workflow and analysis pipelines solutions in the field:

- The analysis pipelines represent recommended or standard workflows general to a field.
- The analysis pipelines are meticulously developed by experts and treated as fixed by the researchers running them.
- They need an execution engine to run, which can require complex installations, configurations, and management. Alternatively, public cloud services can be used, but this requires transferring data to the cloud.
- They are fully automated and do not include human-based WFMS activities.

An example of a more lightweight and lesser-known framework is the *nf-core*

7. <https://openwdl.org/>

8. <https://www.broadinstitute.org/>

9. <https://gatk.broadinstitute.org/>

10. <https://gatk.broadinstitute.org/hc/en-us/sections/360007134212-Computing-Platforms>

11. <https://gatk.broadinstitute.org/hc/en-us/articles/360045944831-DRAGEN-GATK>

12. <https://www.illumina.com/products/by-type/informatics-products/dragen-secondary-analysis.html>

project (<https://nf-co.re/>), which at the time of writing, has 81 available analysis pipelines. Nevertheless, the same properties apply.

The system proposed in this thesis has requirements not targeted by these solutions. Because the composition of tasks and their parameters are unique for each study, the researchers need a convenient way of defining new analysis pipelines, which differs from applying premade, fixed, best practices pipelines. Also, the possibilities for installing custom workflow and pipeline solutions in a secured IT environment are limited. Therefore, the thesis proposes a system allowing the researcher to define a custom analysis pipeline by selecting from a library of configurable tasks. The tasks can be implemented as reusable scripts and woven into a complete analysis pipeline bundled with dependencies. The researcher can download the bundle, transfer it to a secured IT environment having access to the data, and run the analysis pipeline without installing a specialized execution engine. Running the scripts can be combined with the researcher's additional code, e.g., in an R-notebook (Y. Xie, Allaire, and Grolemond 2019). If allowed by the secured IT environment, creating tasks that trigger the execution of other types of pipelines that depend on underlying execution engines is still possible. The tasks then employ such pipelines, e.g., HTS pipelines, as internal components at a lower abstraction level than the overall workflows and analysis pipelines.

Because researchers must formally apply for access to data, the overall study workflow is divided into stages delegated to different roles and requiring human activities to proceed. An example workflow is described in Chapter 9. There are three main stages:

1. Study design
2. Data access application
3. Data processing and analysis

First, the researcher designs the study by following workflow steps such as choosing the type of study, selecting variables and subjects, and composing tasks into a pipeline definition. After the researcher has designed the study, the system can simplify the data access application process by generating lists of needed variables, providing descriptions of the planned data analysis, and providing helpful resources and tools to the researcher. If the person evaluating the application and provisioning data is also a registered user of the system with a data steward role, this stage could potentially include scripts for selecting relevant subjects and other data from the cohort, such as cases and controls, given the criteria specified by the researcher. The system could also coordinate the workflow control between the researcher and the user with the

data steward role. After the data has been made available, the last stage is to generate a runnable pipeline for the remaining data processing and analysis, which the researcher can execute in a secured IT environment.

The three described stages can be mapped to the three functional areas characteristic of a WFMS at the highest level, as identified by Hollingsworth (1995, p. 7). In the terminology of the paper, designing the study and defining the pipeline corresponds to the *build-time process design and definition* and are referred to as *design-time activities* in this thesis. The distribution of work between human actors, seen in conjunction with applying for data access, is the responsibility of the *run-time process control* of the platform. Communication between the platform and human actors and the generation of runnable pipelines are part of the *run-time interactions with users and tools*. For example, the platform builds runnable pipelines by employing external tools and services. However, in this thesis, the actual execution of runnable pipelines is outside the proposed cloud platform's control, because the platform communicates with the user, which subsequently uses the secured IT environment to execute the runnable pipeline.

Lastly, despite being competent in statistics and some scripting, the epidemiology researchers are not system developers. As a result, they rarely have the competence to develop complex software systems that require software engineering practices and an understanding of computer science to achieve acceptable quality. Instead, the researcher should be allowed to focus on the unique parts of their research when defining a workflow. The platform should therefore support a low-code or no-code approach to the study design process. The thesis argues that a natural-language user interface could ease the researcher's work. Paper IV, *Schema-Based Priming of Large Language Model for Data Object Validation Compliance*, proposes a method enabling a natural-language user interface for configuring pipeline definitions and other data object structures.

1.6 Thesis statement

A cloud-based collaboration software as a service (saas) can arguably streamline research by enabling groups to design studies and data analysis pipelines more swiftly; however, the analysis pipelines should be generated as source code in a scripting language already supported by the available secured IT environment to support data processing that complies with the applicable acts, regulations, and ethical guidelines.

1.7 Reading guide

The rest of the thesis is structured into the following parts:

- I) Background
- II) Methods
- III) Toward a Platform for Hybrid Designs and Analysis Pipelines
- IV) Summary of Research
- V) Included Papers

Part I, *Background*, consists of chapters 2–5 and gives the reader an insight into the cross-disciplinary context of the thesis. Chapter 2 is a brief overview of epidemiology. Chapter 3 underscores the importance of bias, associations, and causality in epidemiology. Chapter 4 gives an introduction to four common study designs relevant to the thesis. Chapter 5 briefly covers DNA, genomics, and omics in conjunction with epidemiological studies.

Part II, *Methods*, comprises chapters 6–8 and describes topics crucial to understanding the foundations of the platform proposed in Part III. Chapter 6 gives a general overview of cloud computing. In Chapter 7, a brief overview of WFMS is provided. Chapter 8 explains the concepts of low-code and no-code relevant to the thesis.

Part III, *Toward a Platform for Hybrid Designs and Analysis Pipelines*, proposes a cloud-based solution facilitating the collaboration within research groups using hybrid designs based on existing data from larger cohort studies. First, Chapter 9 introduces a common workflow used to design studies. Next, the overall architecture and design are presented in Chapter 10, followed by a more thorough explanation guiding the reader through the components in Chapter 11. Last, Chapter 12 explains the principles of generating the runnable pipelines.

Part IV, *Summary of Research*, provides a summary of the four papers included in the thesis, their scientific contributions, and the contributions made by the author. An overview is given in chapter 13, followed by a short summary of each paper (chapters 14–17). Chapter 18 gives concluding remarks regarding the work.

Finally, the four thesis papers are included in Part V.

Part I

Background

/2

Epidemiology

Epidemiology is a field that concerns health and diseases in populations. It includes studying diseases and other health-related states, investigating who is affected and in which way, assessing risk factors and causalities, and advising on practices and policies. A widely acknowledged definition of epidemiology is "the study of the distributions and determinants of health-related states and events in specified populations and the application of this study to control of health problems" (Gordis 2014, p. 2). It has also been described as the basic science of community medicine (Strasser 1985). During the COVID-19 pandemic, epidemiology definitively reached the general population's attention. The branch of epidemiology concerned with infectious diseases, including viruses, is known as *infectious disease epidemiology* (Du et al. 2023; Straif-Bourgeois, Ratard, and Kretzschmar 2014). However, epidemiological investigations into non-infectious diseases, such as cancers and cardiovascular diseases, also play a central role, where chemical and environmental exposures, genetics, lifestyle, and socio-economical factors can affect the risk.

Epidemiological studies can broadly be categorized as being *experimental* or *observational* (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 21). In an experimental study, the researcher manipulates one or more variables in an experiment conducted in a controlled environment. A variable can be the treatment given to the study's subjects. For example, the subjects can be randomly assigned to two groups: one receives a treatment, and the other receives a placebo. The researcher can then assess the difference in effect between the groups to establish causality. This is a typical randomized control

trial (RCT) type of experimental study (Alexander et al. 2015d). RCTs are often regarded as the ideal design for studies on treatment effects, but there are limitations (Frieden 2017). For example, RCTs usually do not have sufficient long study periods or population sizes to assess treatment effect durations or identify rare adverse effects (Frieden 2017). Further, many potential experiments would be unethical to conduct. Deliberately exposing individuals to factors increasing the disease risk in an experimental setting is unethical. Instead, observational studies are an alternative. In observational studies, the researchers make observations about outcomes and exposures under investigation while avoiding affecting any variables, measured and unmeasured, or altering the environment of the individuals in the study. Observational studies are either *descriptive* or *analytical* (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 21). Descriptive studies report the distribution and characteristics of health-related events in populations, often focusing on the "Who, What, Where, and When" questions, describing the patterns of health outcomes in terms of people, places, and time. On the other hand, analytical epidemiology seeks to assess risks and causality, usually by comparing groups regarding differences in exposure to potential risk factors or interventions.

The various types of studies can be placed in a hierarchy according to the strength of the evidence, called an evidence ladder or pyramid (Cooper 2016). There are several versions of the ladder, with some variations. Study types far up the ladder are presumed to have a higher strength of evidence than those far down. However, here it is important to remember that the quality of the studies can vary regardless of where in the hierarchy they are located (Cooper 2016). Investigations usually start at the bottom and then move step by step up the ladder (Gordis 2014, pp. 244–245). Case studies are close to the bottom, often consisting of only one or a few cases. This type of study can consist of a medical doctor making clinical observations at the bedside and describing a condition or disease picture. This, in turn, can lead to new studies, often observational. As the knowledge and hypotheses become more extensive because of the studies, promising treatments can be developed and assessed using experimental studies such as RCTs. Even further up the ladder are meta-studies and systematic reviews, which summarize research in the field from several sources.

/ 3

Bias, Associations, and Causality

Bias requires special attention in epidemiology. It is easy to introduce systematic errors in the design, execution, or analysis of a study, which in turn leads to skewed and invalid results. Two main categories of bias are selection bias and information bias. Selection bias applies to systematic errors in how participants are selected and grouped, whereas information bias applies to errors in information gathering. One type of information bias is misclassification bias, which occurs, for example, if participants with a disease mistakenly are classified as healthy. Recall bias is another form of information bias, which can happen if the participants misremember the past. By using established study designs (Chapter 4) epidemiologists can better understand and reduce the risk of particular kinds of bias.

An early catalog of biases was published in the 1979 paper *Bias in Analytic Research* (Sackett 1979). The paper discusses the importance of bias in research and recommends the continued development of an annotated catalog. Today, a bias catalog supported by the Centre for Evidence-Based Medicine (CEBM) and the University of Oxford is available online (<https://catalogofbias.org/>).

It is essential to understand the difference between observed *associations* between factors and *causes* (etiology) of a health outcome. Factors statistically

associated (correlated) with one another do not necessarily imply causal relationships. Modifying one of the factors may not even affect the other, for example, if the association is due to an underlying confounding factor affecting both (Alexander et al. 2015a,b). In epidemiology, various methods are used to control for possible confounding factors. Another example is *reverse causation* (Porta 2014, p. 249), which is mistaking cause for effect and vice versa. The Bradford Hill criteria are commonly used to assess causality (Hill 1965), but some variations exist (Gordis 2014, pp. 250–260).

/4

Study Designs

In analytical epidemiology, researchers use standard *study designs* as a foundation for designing studies and avoiding typical systematic errors when assessing relationships between treatments or exposures and health outcomes. Each design has its well-known strengths and weaknesses. Deciding on an appropriate study design is essential because many serious problems and mistakes at the design stage of a study cannot easily be rectified in subsequent stages (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 21). This chapter describes a selection of relevant study designs.

4.1 Case-Control Studies

Case-control is a much-used study design in which participants are grouped by having experienced the studied health outcome or not. A participant with the health outcome is called a *case*, and a participant without is called a *control*. The researchers will pair one or more controls with a case. The controls should be similar enough to their case to be comparable. The next step is to investigate suspected risk factors or exposures. The participants will ordinarily answer questionnaires about the degree of exposure to risk factors. Because the questions are retrospective, case-control studies are prone to recall bias. Selecting cases and controls without introducing selection bias can be challenging (Wacholder et al. 1992a,b,c). For example, the two groups can be systematically different from each other without it being obvious. However, case-control stud-

ies also have appealing advantages, e.g., case-control studies require relatively few participants and little time before results can be analyzed.

4.2 Longitudinal Studies and Prospective Cohorts

Longitudinal studies follow subjects over time with continuous or repeated monitoring of risk factors or health outcomes, or both (Coggon, D. Barker, and Rose 2009). The studies are additionally population-based if they aim to be generalizable to the whole population addressed in the study hypothesis, not only to the individuals included in the study (Lieb 2013).

A prospective cohort study is a longitudinal study that only recruits individuals who do not have the investigated health outcome. The participants answer questionnaires regarding exposure to suspected risk factors, and they can undergo relevant medical examinations, donate blood, and so on. The participants are then followed up over a specified period. For diseases such as cancer, a certain percentage of the participants will develop cancer annually. As time passes, if the cohort is sufficiently large and the period long enough, more participants will probably be diagnosed with cancer. At the end of the study, the researchers examine the difference in cancer incidences for exposed and non-exposed groups—or groups with different degrees of exposure to a suspected risk factor. One advantage of prospective cohort studies is that this study design reduces the problem of recall bias, which is a challenge in case-control studies. The reason is that questionnaires, surveys, and sample acquisitions occur in advance before knowing who will develop the health outcome in the future. However, there are downsides. The studies usually required far more participants than case-control studies, especially for rare health outcomes. The recruitment and follow-up process becomes complicated, and the number of samples that need to be analyzed makes the projects costly. Also, the studies may have to run for several years before researchers can analyze the results.

4.3 Hybrid Designs

Some of the problems with case-control and cohort designs can be reduced by combining the two types in a hybrid design. Two popular types of hybrid designs are nested case-control and case-cohort (Kleinbaum, Sullivan, and N. D. Barker 2007, p. 39). Both are case-control studies inside of cohort studies. Hybrid designs are well suited for large population-based prospective cohort

studies. In this type of cohort study, a wide range of exposures is surveyed for each participant. The intent is to collect a broad range of information that can be interesting later without this being known in advance. Blood and other biological samples can be stored unanalyzed in biobanks. The cohort can then be used as a starting point for selecting cases and controls in other, smaller studies. Such hybrid designs have several advantages. It is more cost-effective to analyze samples only when needed in a smaller case-control study. You can also perform types of analyzes that were unforeseen when the study started. The risk of recall bias is reduced for the same reasons as in an ordinary cohort study. The risk of selection bias is also reduced because cases and controls are selected from the same population. This is in contrast to simple case-control studies, where the cases and controls often are selected from hospital departments or other places where the groups are already systematically different from each other. Hybrid study designs can thus provide the best of both worlds.

4.4 Cross-Sectional Design

The last type of study design covered in this chapter is *cross-sectional*, also known as prevalence studies. A cross-sectional study is a snapshot of the population at a particular time. It does not include information about when the health outcome occurred. Information about exposure is included. A problem with this type of study design is that temporal relationships are missing, i.e., no information about the timeline of events. Therefore, it could be that exposure to a suspected risk factor happened after the diagnosis. The risk factor could then instead be a result of the health outcome. And if the health outcome is a potentially lethal condition or disease, then what appears to be a risk factor may, in reality, be a survival factor found in people who live longer with a disease or condition—a type of bias called survival bias. Still, there are situations where cross-sectional designs are justified.

/5

DNA, Genomics, and Omics

DNA is a double-stranded molecule in human cells that carries our genetic information in sequences of four nucleotides (adenine, cytosine, guanine, and thymine) (Miller 2022a). This genetic information is inherited from our parents and guides the production of proteins from amino acids, among other functions. A gene is a segment of DNA that codes for a specific function, such as a protein, and there are about 20,000 genes in the human genome (Miller 2022b). Occasionally, mutations can alter the DNA sequence. Inherited mutations are called constitutional or germline and affect all cells, whereas somatic mutations are non-inheritable and affect some cells. Somatic mutations are often a result of environmental factors or diseases (Frost 2022).

The Human Genome Project (<https://www.genome.gov/human-genome-project>) completed history's first sequence of the human genome in April 2003. Today, high-throughput sequencing (HTS), also called Next-generation sequencing (NGS), are technologies that allow for the rapid sequencing of large amounts of DNA or RNA at a significantly reduced cost. The HTS technologies perform sequencing of millions of small fragments of DNA in parallel, and it is now possible to sequence the entire human genome in as little as a day (Behjati and Tarpey 2013).

Genomics is relevant to epidemiology as inherited genetic variations can influence health risks. Genome-wide association studies (GWAS) analyze genomes for genetic variations, aided by software such as the Broad Institute's Genome Analysis Toolkit (GATK). Three main types of genetic variations are analyzed

in GWAS: Single-nucleotide polymorphisms (SNPs), which are changes in a single base-pair; Inserts and deletions (INDELS), which are variations of 1 to 10,000 base pairs in a DNA sequence; and copy-number variations (CNVs), which are changes in the number of repeated sequences of base pairs.

Omics refers to fields in molecular biology that study the genome and its products, including genomics, epigenomics, transcriptomics, proteomics, and metabolomics (Vailati-Riboni, Palombo, and Loo 2017). Genomics focuses on DNA and genes, while epigenomics studies gene accessibility and regulation through DNA packaging. Transcriptomics involves the study of RNA, which carries information from genes for protein synthesis. Proteomics examines an organism's set of proteins, which have various roles, such as enzymes, structural components, and messengers. Metabolomics deals with the body's chemical processes and the molecules involved in metabolism, including proteins, lipids, carbohydrates, and nucleotides. Lastly, multi-omics combines data from various omics fields.

Omics are relevant to modern epidemiology. For example, epigenomics and transcriptomics can, in some cases, reveal biological markers for diseases in blood or tissue.

Part II

Methods

/6

Cloud Computing

Cloud computing is a type of utility computing. It provides computing resources analogously to how utility companies provide electricity, water, and gas services. Instead of users and organizations owning and managing all needed physical computer resources on their premises, a shared pool of resources in data centers elsewhere provides the computational power, storage, network, applications, and services. A third party typically owns the cloud, and the services are accessible over a standard network and consumed on a pay-per-use basis.

The National Institute of Standards and Technology (NIST) has formulated a widely used definition of cloud computing, which starts with: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models" (Mell and Grance 2011). Next, the definition continues with listing and describing each of the characteristics, service models, and deployment models.

The four cloud deployment models are public, private, community, or hybrid. *Public clouds* are for open use by the general public but exist on the premises of the cloud providers. In terms of market share, the four largest public cloud providers in 2023 are Amazon Web Services (AWS), Microsoft Azure (AZURE), Google Cloud Platform (GCP), and Alibaba Cloud (Alibaba) (Richter 2023).

On the other hand, *community clouds* are for specific consumer communities with shared concerns, such as special compliance considerations and security requirements. HUNT Cloud is an example of a community cloud offered by the Norwegian University of Science and Technology (NTNU). It specializes in solutions for scientific cloud computing and has an extensive compliance framework to protect data privacy.

The three service models listed in the NIST definition are:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS).

For this thesis, SaaS is the most relevant of the three. This service model lets consumers use a provider's applications on a cloud infrastructure. The consumers access the applications using thin clients (e.g., web browsers) or application programming interfaces (APIs), but the SaaS consumers do not manage or control the underlying cloud-based platform or infrastructure. In the thesis, the consumers are the research groups, whereas the application provider is the part offering the proposed collaboration software to the research groups as a service running in a public cloud. The public cloud itself is owned and managed by a third-party cloud provider.

The term multi-tenant means that several tenants, or consumers, share the same resources while being isolated in separate environments and precluded from knowing about or accessing the other tenants' environments. The cloud is, per (NIST's) definition, inherently multi-tenant through its use of resource pooling, serving a plethora of consumers sharing the same physical resources in a manner giving a sense of location independence. The term is also relevant to SaaS. A SaaS can follow a single, mixed, or multi-tenant model (<https://learn.microsoft.com/en-us/azure/architecture/isv/application-tenancy>). In a single-tenant model, a separate software installation, or instance of the software, is deployed for each consumer. For a multi-tenant solution, a single instance of the application is shared by multiple consumers, but each has an isolated environment where its data and configurations are inaccessible to the others. In a mixed model, specific layers or tiers of the application are deployed per consumer, and the rest is shared. For example, each consumer can have a separate database (DB) instance but share the same instance for the rest of the application. From a SaaS provider's perspective, a multi-tenant solution is cost-effective due to resource utilization. Also, the management of the SaaS platform may become more straightforward due to centralization. Often, single and mixed-tenant solutions are used to ensure better performance through

provisioning resources separately per consumer or because security requirements demand separate DBs. For the SAAS proposed in the thesis, neither the computational intensities nor user traffic are expected to be particularly demanding. The computations for the data analysis pipelines are separate from the public clouds. Further, none of the regulated data are stored or processed in the public cloud. Consequently, a multi-tenant solution is possible.



Workflow Management Systems

This thesis defines a workflow as a repeatable series of steps in a work process, organized in a specified order from start to end, possibly including conditional branches, where each step includes one or more activities to execute before proceeding to the next. The steps can involve a combination of human and machine-based activities.

A workflow management system (WFMS) is a software system that facilitates the execution of workflows by enabling the computerized definition and management of workflows and automating the coordination of human and machine-based activities. A closely related term is business process management system (BPMS) and is sometimes used synonymously with WFMS in literature, which is the case for the relevant systematic review on BPMS architectures by Pourmirza et al. (2017).

A classical WFMS architecture reference model is found in Hollingsworth (1995). It describes that, at the highest level, all WFMS solutions have functionality for:

1. Defining workflows at build-time
2. Managing the progress of the workflows at run-time

3. Interacting with human users and IT application tools for processing the various activity steps at run-time

In the reference architecture, the user defines workflows using a build-time definition tool. The definition tool produces a *process definition* that a workflow management (WFM) engine manages and interprets at run-time. The WFM engine then orchestrates or choreographs the activities of users and IT application tools, often by using a combination of internal and external third-party system components or services. In contemporary literature, the terms *orchestration execution engine* and *choreography execution engine* can be seen used for WFM engines (Pourmirza et al. 2017). Generally, there is a distinction between orchestration and choreography when describing architectures involving services—e.g., *microservices* (IBM Cloud Team 2021). Orchestration implies that a centralized authority coordinates service interactions, whereas choreography usually implies an event-based architecture with a more distributed and autonomous interaction model. A case for using a workflow engine to orchestrate microservices instead of relying on choreography is made in Nadeem and Malik (2022).

All the large cloud providers offer WFMS as part of their standard portfolio. Table 7.1 lists examples of WFMS services offered by the four largest cloud providers.

Table 7.1: Cloud providers and WFMS service

| | | |
|---------|-----------------------|---|
| AWS | Simple Workflow (SWF) | https://aws.amazon.com/swf/ |
| GCP | Workflows | https://cloud.google.com/workflows/ |
| Azure | Logic Apps | https://learn.microsoft.com/azure/logic-apps/ |
| Alibaba | Serverless Workflow | https://alibabacloud.com/product/serverless-workflow/ |

/ 8

Low-Code and No-Code Development

Richardson et al. (2014) coined the term *low-code* in a Forrester trend report about the emergence of new development platforms for customer-facing applications, dramatically reducing the need for hand-coding. However, no uniformly accepted definition of low-code exists. The systematic review of low-code and usability by Pinho, Aguiar, and Amaral (2023) addresses this and defines low-code as:

Low-code development is a set of approaches, technologies, and tools that enable rapid application development through techniques that reduce the amount of code written.

These approaches can make it possible for end-user developers to program software and use techniques and tools, often including but not limited to model-driven engineering, domain-specific languages, and drag-and-drop mechanisms.

A closely related term is *no-code*, which aims to remove the hand-coding altogether. Tracing the term's origin is difficult, but an early usage example is the website NoCode (<https://nocode.tech>), which has provided a curated directory of tools for non-technical entrepreneurs since November 2016. A May 2017 snapshot of the website by the Internet Archive Wayback Machine

(<https://web.archive.org/>) shows that the directory included categories such as website builders, app-builders, customer support tools, forms and survey creators, and more (<https://web.archive.org/web/20170503174122/https://www.nocode.tech/>). The site's category descriptions align with what we today consider to be the aims of no-code, such as making tedious and time-consuming development tasks quick and easy without requiring code or design knowledge.

A search for published papers with "no-code" or "no code" in the title or abstract was conducted using Web of Science, ACM Digital Library, and IEEE Xplore, and the earliest found published academic work using no-code in its current meaning was Fryling (2019). The work was a brief conference workshop description, and from the text, it is clear that the term was already in established use. It also ties no-code to model-driven development, described further down in this chapter.

Development tools fitting into the categories of low-code and no-code have existed far longer than the terms themselves. For example, the 1992 paper by Baldassari-Mather (1992) presents a visual design tool, Hewlett Packard's HP VEE-Test, using a flow-chart style of design for creating automated test programs without the need for traditional programming languages. Thus, low-code and no-code are categories that emerged after a decades-long evolution of software tools aiming to make development more time-saving, cost-efficient, and easily accessible to non-programmer users. Typically, the tool's target users—frequently referred to as *line-of-business users*—are assumed to be relatively inexperienced in programming but proficient in their respective fields.

The vendors considered to be the first movers in modern low-code development tools originated from three partly overlapping neighborhoods of the software industry (Richardson et al. 2014):

- Business process (BPM) and case management
- Web content platforms
- App platforms

The vendors listed under BPM and case management in Richardson et al. (2014) were all reported to share strengths in the areas of BPM, case management, and workflow, which shows that workflows have played an important role in low-code and no-code since the early years.

Under the hood, many of the methods used to enable low-code and no-code are related to model-driven development (Fryling 2019; Pinho, Aguiar, and Amaral

2023). Following (Pinho, Aguiar, and Amaral 2023), model-driven development (MDD) is a software development methodology based on software-defined *models* that represent simplifications or abstractions of a problem domain. The models can be designed and generated using tools and transformed into software using model transformations. Also, *metamodels* can be employed to define a modeling language and ensure that models comply with the language definition. MDD can be used as a foundation for automated software development, such as low-code and no-code.

Brambilla, Cabot, and Wimmer (2012, pp. 26–33) identifies source code generation and model interpretation as two central approaches in automated software development. In the typical case, multiple model-to-model transformations are applied before a final step produces an executable model either in source code form or interpretable by an execution engine. This thesis takes a source code generation approach and generates scripts for runnable pipelines from a pipeline definition model, which the user designs using a low-code or no-code approach. In turn, the pipeline definitions follows the rules of a metamodel defining an overall structure applying to analysis pipelines for the chosen type of study design. The runnable pipelines described throughout this thesis corresponds to the executable models described in Brambilla, Cabot, and Wimmer (2012, pp. 28–31) that are generated by utilizing a rule-based template engine, i.e., the code generator consists of a set of templates with placeholders that once applied (instantiated) on the elements in the model, produce the code. A systematic review of template-based code generation is found in Syriani, Luhunu, and Sahraoui (2018).

Part III

Toward a Platform for Hybrid Designs and Analysis Pipelines

/9

A Workflow for Designing Hybrid Studies and Data Analysis Pipelines

At the heart of the proposed platform is the research group’s workflow for designing studies and analysis pipelines using subsets of the same regulated data as a basis for numerous studies. This chapter presents the overall structure of a study design workflow for hybrid designs, for example, case-control studies nested in the same population-based prospective or longitudinal cohort study.

Figure 9.1 illustrates that the overall workflow has three basic steps. The first step comprises design-time activities for defining the study design, variables, selection of subjects (participants), and data processing and analysis-related tasks for the study. The next step is to apply for data, which requires information from the first step. After data provisioning, the final step is to generate a runnable representation of the data processing and analysis tasks—a runnable pipeline—that can be transferred and executed in the secured IT environment where the provisioned data are available.

In Figure 9.2, the overall workflow has been expanded into finer-grained steps. The expanded version is to be understood as an example because the system should enable different research groups to define and evolve overall workflow

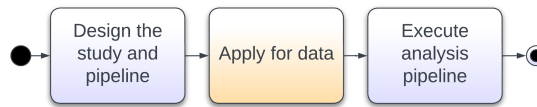


Figure 9.1: The figure illustrates the three basic steps of the overall workflow.

structures in alignment with their internally used methods. The example is based on a combination of the already described needs, hybrid designs in general, and the design used for the nested case-control study in Agerbo, Nordentoft, and Mortensen (2002).

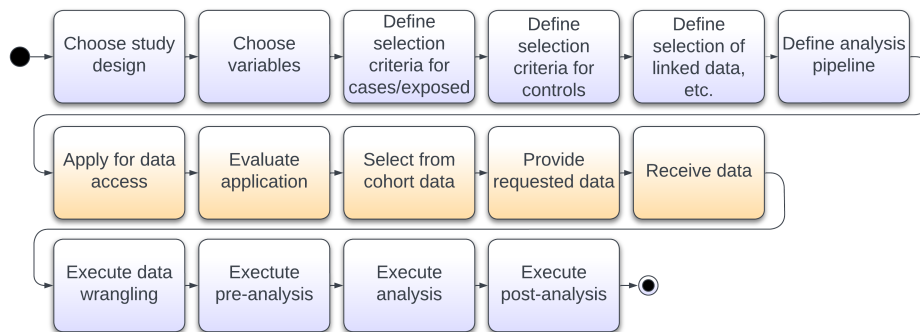


Figure 9.2: The figure illustrates an expanded version of the overall workflow.

9.1 Designing the study and pipeline

The first row of the expanded workflow is a sequence of design-time activities for defining the study design, variables, subject selection, and data processing and analysis-related tasks for the study, described in this section.

The researcher first selects one of the predefined study designs, e.g., a nested case-control or case-cohort study.

The next step is to select the variables needed for the study, including the exposure, outcome, and control variables. Because there can be thousands of available variables, the system should provide a search tool utilizing a data dictionary of variable meta-data. A searchable database of variables and categories for Norwegian data sources is already present at Helsedata (<https://helsedata.no/>). For example, for NOWAC, Helsedata has meta-data on 12,105 variables from questionnaires and registers (*Variabler 2023*) grouped in 31 categories. Additionally, the researcher may need variables derived from other variables, e.g., index variables. Here, the system could utilize meta-data about

registered wrangling functions' input variables to generate a list of the needed source variables for computing the derived variable, if available.

After choosing the variables, the researcher can specify the criteria for selecting cases, exposed, or treated subjects, depending on the type of study—and afterward, the selection criteria for the study's controls and possibly matching method. For some studies, data linkages may also be needed.

Also included in the design-time activities is the design of the analysis pipeline part of the workflow that can be converted to a runnable pipeline and run in a secured IT environment after data provisioning, described in Section 9.3.

9.2 Applying for data

The second row of workflow steps illustrated in Figure 9.2 represents activities related to applying for access to data.

Usually, the application must include a list of all needed variables, possibly selection criteria, and a description of the study's objective and methods. The system can help the researcher in the application process by generating information needed for the application from the pipeline definition. It could additionally provide other resources, such as relevant documents and application forms.

The subsequent steps depend on whether the person processing the application is a registered user of the system with a data steward role. Often, a research group member is responsible for evaluating the application, as well as extracting and provisioning the requested cohort data subset. In that case, the system could alert and give control of the data access application steps to this person. The applicant could then follow the progress through status messages.

If the evaluation results in approval, the system could help the data steward extract the data requested by the applicant by providing scripts for selecting data based on the pipeline definition.

After providing the data, the applicant can be alerted.

9.3 Executing the analysis pipeline

The third row of Figure 9.2 illustrates workflow steps for data processing and analysis tasks, i.e., the data analysis pipeline. At this stage of the overall workflow, the researcher can access data. The system can generate a runnable pipeline from the pipeline definition, which the researcher transfers to a secured IT environment and runs.

In the wrangling step, data can be prepared for further processing, for example, by applying scripts for computing index variables from a selection of source variables. After wrangling comes a pre-analysis step for processing data. Examples of pre-analysis tasks are automatic stratification or balancing of the sample by pruning to reduce bias. The method presented in Paper II, *Autostrata: Improved Automatic Stratification for Coarsened Exact Matching*, is an example of a method applied in a pre-analysis step. The next step is composed of data analysis tasks, such as regression modeling and estimating risk. Paper III, *Greedy Knot Selection Algorithm for Restricted Cubic Spline Regression*, is an example of a method belonging to the analysis step. Lastly, an optional post-processing step can facilitate evaluation, interpretation, and documentation-related activities.

/10

Architecture and Design

This chapter presents the core system architecture of the proposed platform, which can be embedded in a more comprehensive saas solution offering a myriad of helpful collaboration features. Further, many existing cloud-based solutions can easily integrate as components of this platform. However, the chapter focuses on the architecture related to the contributions of the thesis instead of standard features provided by existing integrable solutions.

The system architecture spans three distinct computing environments, as illustrated by Figure 10.1—the public cloud, the user’s local computer, and the secured IT environment. In the architecture, the user’s local environment sits between the public cloud and the secured IT environment and communicates directly with these, but there is no direct communication between the public cloud and the secured IT environment.

The data provisioned for analysis are accessible only within the secured IT environment. To analyze the data, the user typically logs in to a virtual machine (VM) (Barham et al. 2003; Sugerman, Venkitachalam, and Lim 2001) in the secured environment, where a remote directory containing the data files has been mounted into the filesystem. A selection of preinstalled software can be used to analyze data, and a scripting language and integrated development environment (IDE) for statistical analysis can be assumed to be available, such as R-studio. For security reasons, the user can generally upload but not download files, and the VM deliberately does not have internet access. Despite the strict limitations, the user can still upload scripts and run in the VM, a



Figure 10.1: The figure illustrates the system’s three primary environments: The cloud, the researcher’s local computer, and the secured IT environment with a VM having access to the provisioned data for the study.

central reason for the proposed system’s architecture.

Another aim of the system is to save time and resources within a group by encouraging collaboration and enabling sharing of custom functions developed for a particular study in a form reusable in subsequent studies. However, a secured IT environment is inappropriate for hosting the platform because of its numerous restrictions due to security policies. Consequently, the architecture is split into one part for the functionality requiring access to regulated data and another independent of this data. An on-premise or cloud-based system is possible for the latter because no sensitive data is involved. Here, a cloud-based saas platform is advantageous over an on-premise system, as explained in Chapter 6 on cloud computing.

For the rest of the thesis, it is understood that the system is available to research groups as a multi-tenant saas solution where different groups share the same physical installation of the platform hosted on a cloud infrastructure. Each group shares the same underlying installation. However, each group has its own separate and isolated environment, which includes configurations, data, and user management, and one group cannot access another’s data or configurations. The groups’ environments are customizable to their specific preferences. Employing a multi-tenant approach allows for efficient use of resources and simplified maintenance and updates, allowing changes to be made to the core system without affecting each group’s customized settings and data.

The system has several user roles, of which Figure 10.2 shows the five most relevant to the described part of the architecture. A user can have more than one role. The platform administrators have responsibilities that involve the management of user and group accounts, access rights, system maintenance, updates, and monitoring system status. The Platform Admin role is independent of any research group memberships and likely should not be given to

group members to avoid conflicts of interest between groups using the platform. On the other hand, the Group Admin role is held by group members. Users having this role manage the group's SAAS account, its users, and their roles. The group administrator also has access rights allowing management of preferences, customizations, and configurations according to the group's needs and preferences and other intragroup administration activities. The Function Contributor role is for users who write and share functions that the group's researchers can choose when designing a data pipeline for a study. The Data Steward role is for users who can evaluate applications for access to regulated data and who can provision the data. If such group members exist, handling applications can be supported by being integrated into the workflow. Lastly, the Researcher role is for users who use the system to design and conduct studies.

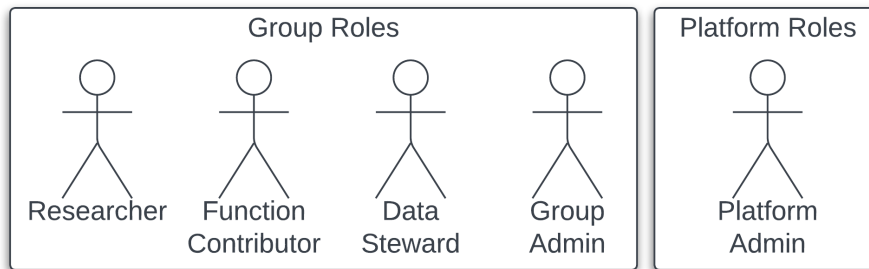


Figure 10.2: Five different user roles in the presented architecture

Figure 10.3 is a UML component diagram (Larman 2002, p. 605) illustrating a high-level view of the components of the system's architecture involved when a user with a researcher role designs a study by following the workflow. The diagram organizes the components into two subsystems, *Study Workflow Client* and *Study Workflow Services*. The former is the researcher's client for designing the study; it can, for example, be a single-page application (SPA) (Fink and Flatow 2014) running in a browser on the researcher's local computer. The latter is composed of services (Kleppman 2017, pp. 131–133) running in the public cloud.

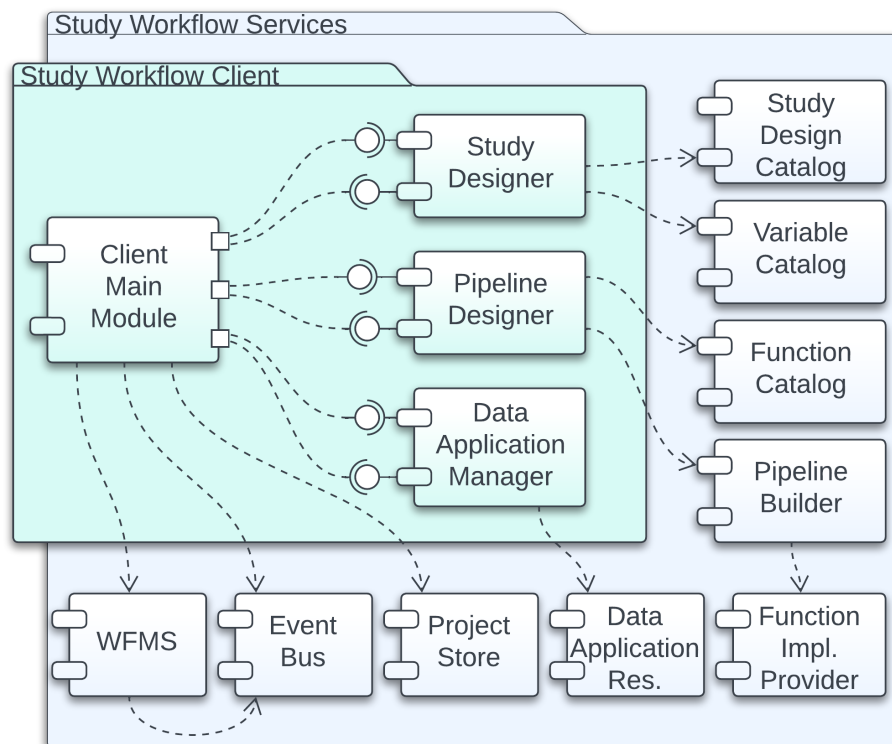


Figure 10.3: The figure is a UML component diagram giving a high-level view of the components in the architecture relevant to the study design workflow as used by a researcher role.

/ 11

Study Workflow Use Case

The components of the architecture illustrated in Figure 10.3 are now explained in a writing style informed by casual use cases (Cockburn 2000) with additional explanations of components. In this use case, the researcher has opened the client in order to design a new study. The `Client Main Module` creates a new empty project for the workflow and stores it using the `Project Store`. Next, the main module calls the `WFMS` to register a new workflow in its initial state. The main module both exposes and requires interfaces from the other client modules, the `Study Designer`, the `Pipeline Designer`, and `Data Application Manager`, enabling the main module to control the others, as well as allowing these to make requests back to the main module in an orderly manner. Interfaces also make the separation of concerns clear, making different implementations of the components interchangeable. After initializing the new project, the main module opens the study designer. The study design module includes the user interface (UI) for designing the study, up to the analysis pipeline design process. The main client module publishes an event via an `Event Bus` (Tanenbaum and Van Steen 2014, pp. 35–36) informing that the workflow has entered the study design phase. The `WFMS` picks up and processes the event.

11.1 Study Designer

The Study Designer lets the researcher choose a predefined type of study design from the Study Design Catalog, where typical examples are nested case-control or case-cohort. A group administrator can add study designs to the catalog so that the exact content can vary from group to group. When the researcher chooses a study design, a data structure representing the study is created. The structure contains the steps and values for a workflow corresponding to the first row of the expanded workflow illustrated in Figure 9.2, except for the analysis pipeline design step. The designer module manages this structure and provides the UI for guiding the researcher through the steps. To persist and retrieve the structure, the designer calls functions on the main module's exposed interface, and the main module calls the Project Store service. As part of designing the study, the researcher must specify which variables are needed from the potentially large number of available variables. The study designer lets the user search for information about variables by employing the Variable Catalog service. After the researcher has finished the study design workflow, the study designer asks the client's main module to persist the data in the project storage and update the state of the overall workflow. The main module updates the workflow state by publishing an event that the WFMS processes.

11.2 Pipeline Designer

Next, the Client Main Module switches from the Study Designer to the Pipeline Designer module. The pipeline designer gets the data structure created by the study designer via the main module. This structure contains information on the researcher's chosen variables and other parameters the pipeline designer can use. The pipeline designer also creates a *pipeline definition* structure, which will hold the information needed to build a `runnable pipeline` later. As with the study design structure, the Project Data service provides persistent storage, which the pipeline designer uses via the main module's provided interface. The researcher defines the pipeline by using a no-code or low-code UI, possibly with a natural-language user interface (NLUI), assisting the researcher in choosing from functions from the Function Catalog. The Function Catalog provides meta-data about available functions, including required and optional parameters, datatypes, permitted values, and documentation. Paper IV in the thesis, *Schema-Based Priming of Large Language Model for Data Object Validation Compliance*, presents an approach letting a Large Language Model (LLM) be used to assist a researcher. Features similar to GPT-4 *function calling* (<https://platform.openai.com/docs/guides/gpt/function-calling>) could be employed as well. However, these implementation details

are mentioned to highlight possible solutions and are not essential to the architecture.

When the researcher has completed defining the analysis pipeline, the `Pipeline Designer` can pass the pipeline definition to the `Pipeline Builder` service. This service interprets the pipeline definition and generates files (e.g., script files) with the actual implementation of the analysis pipeline. The function implementations are retrieved from the `Function Implementation Provider` service, which is the service involved when a `Function Contributor` adds new reusable functions for sharing with the group. It also provides the dependencies needed by the functions. The functions, e.g., scripts, are weaved into a pipeline by correctly connecting the functions in order and ensuring that function arguments are passed as specified by the pipeline definition. The pipeline builder bundles the files and the dependencies needed to run the pipeline in the secured IT environment into an archive. Chapter 12 explains how runnable pipelines based on scripts can be built using a template engine.

Finally, the researcher can download the runnable pipeline bundle from the cloud to the local environment and transfer it to the VM in the secured IT environment. In the VM, the bundle can, for example, be unpacked and imported into the IDE, where the researcher can run it. Researchers often use a notebook environment allowing the combination of text and interactive execution of code chunks.

11.3 Data Application Manager

Data must be available before the researcher can run the data analysis pipeline. The researcher must send an application to the data steward who evaluates the application and provides the requested data. In the system's architecture, the `Data Application Manager` module provides tools and resources to help the researcher to apply. If the data steward is a group user with the `Data Steward` role, the system can contain workflow steps for more efficient communication between the researcher and data steward during the application process. The status of the application can be tracked, and changes can be signaled between running instances of the client. When the status of the application is updated, the application manager module calls the main client module, which triggers the WFMS. After the WFMS updates the application status in the workflow, it publishes an event on the event bus. The running clients will then pick up and process the event and can immediately update the UI accordingly. It is also possible for WFMS to automatically send messages to researchers and data stewards by mail or other channels as part of the application process. If the data steward provisions data, the data will be made available within

the secured IT environment, and the researcher will be notified by the WFMS. However, the data will never be transferred through the public cloud or the researcher's local environment.

/12

Building Runnable Pipelines

In the key scenario described in Section 1.2, the researcher uses scripts to analyze the study data in an instance of R-studio running on the VM in the secured IT environment. This chapter explains how the proposed system can build scripts from pipeline definitions. Script files are plain text files with source code typically interpreted by a scripting engine at runtime. Because the source code is text, it is possible to use a template engine for assembling a runnable pipeline in source code format from a pipeline definition data object. This approach is informed by model-driven development (MDD) and template-based code generation (Chapter 8).

Paper IV shows how schemas can document and define the rules for a pipeline's possible steps and functions. The schema defines where a function is valid in the pipeline, its required and optional parameters, the parameter datatypes, and value restrictions. When a researcher defines a pipeline, it is represented internally in the system as a data object that complies with the schema. This pipeline definition data object is serializable to a suitable text-based representation. Examples of standard text-based data formats are *XML*, *JSON*, and *YAML* (Edelman, Lowe, and Oswalt 2018, pp. 151–176). Such serialized pipeline definitions are passed to the `Pipeline Builder`, which is responsible for creating executable scripts bundled with dependencies and resource files.

The templates contain a combination of source code written in the scripting language and placeholder expressions written in the templating language. There is a main template representing the general structure of the pipeline, and subtemplates for calling the functions chosen by the researcher are inserted dynamically into this template as specified by the placeholder expressions. Further, the main template implements a way of calling functions in the correct order and gluing the function calls and responses together into a pipeline. The `Pipeline Builder` parses the serialized pipeline definition and passes the resulting data object and the templates to the template engine to assemble and generate the final script.

More detailed, the pipeline definition data object holds the values for the researcher's chosen pipeline functions and parameters. The actual function implementations are included in *partial templates* that the template engine injects into the pipeline structure defined in the main template. These partial templates also include expressions for injecting values from the pipeline definition data object, such as function arguments originally decided by the researcher during the design phase. The `Pipeline Builder` retrieves the needed function implementation partial templates from the `Function Impl. Service`. The `Function Impl. Service` is also responsible for providing the `Pipeline Builder` with any additional files and resources that a function needs.

An example of a simple yet flexible templating language is *Handlebars* (<https://handlebarsjs.com/>). It is compatible with the *Mustache* (<http://mustache.github.io/>) templating language but includes extra features valuable when dynamically assembling scripts from multiple partial templates. Implementations exist for a wide range of programming languages, such as JavaScript¹, Java², .NET³, and Python⁴.

1. <https://handlebarsjs.com/>
2. <https://github.com/jknack/handlebars.java>
3. <https://github.com/Handlebars-Net/Handlebars.Net>
4. <https://github.com/wbond/pybars3>

Part IV

Summary of Research

/ 13

Overview

The proposed cloud-based platform has now been described. The following part of the thesis gives an overview of the included papers and how they fit into the overall thesis context. In this connection, it may be worthwhile to refresh Figure 1.1. The figure is a concept map showing topics in the thesis and included papers. Furthermore, it illustrates four partially overlapping areas with gray tones as a background, representing topics related to Paper I, Paper II & III, Paper IV, and the proposed platform in the thesis.

Paper I covers a relatively broad area that includes topics in epidemiology, study designs, workflow, analysis pipelines, and analytics platforms. Papers II and III are related to statistical methods that fit into different steps of the analysis pipelines; Paper II involves balancing samples or matching cases with controls to adjust for confounding factors as a pre-analysis step, whereas Paper III is about curve regressions commonly used in epidemiology and fits into the analysis step.

The figure illustrates that no papers cover the wrangling step. However, this is covered instead by a report on wrangling smoking variables in NOWAC and the associated R-scripts (<https://github.com/jo-inge-arnes/smoking-variables>).

Paper IV shows that a UI supporting natural language may be used to design analysis pipelines in low- or no-code settings.

Finally, the figure illustrates that excluding the papers, the thesis has topics distinct from the papers, focusing on a cloud platform for collaboration, challenges around regulated data, the required use of a secured IT environment, and code generation as a solution.

/ 14

Paper I

The Beauty of Complex Designs

The Beauty of Complex Designs is the first paper in the thesis and establishes the Ph.D. project's initial context. As an independent work, it is a position paper having a threefold objective:

1. To provide an overview of systems epidemiology in the setting of the NOWAC study
2. To describe how novel systems epidemiological studies can be designed by using data from population-based prospective cohorts
3. To identify computer science research opportunities, discuss requirements, and suggest a systems architecture for swiftly designing and exploring nested studies

The paper starts with a historical account of the developments in analytical observational epidemiology leading up to the introduction of systems epidemiology in 2008 (Lund and Dumeaux 2008) and the processual approach in 2015 (Lund, Plancade, et al. 2015). After the introduction, the paper gives an overview of the NOWAC study and data, which is usable for hybrid systems epidemiological study designs. Next follows a description of how studies can be designed using data from NOWAC, including an example of the design process. Finally, after showing that combining data in numerous ways to design novel

studies is possible, the paper argues that a computer system enabling a swift and uniform process for this purpose would be beneficial. From this position, an architecture is proposed based on the requirements of such a system. However, the architecture presented in the thesis differs considerably from the earlier work found in *The Beauty of Complex Designs*.

As a small note, the paper states that the number of women included in the NOWAC study is 172,556. This number is obtained from a figure published at the official NOWAC website by summing the number of recruited participants per first-time questionnaire, shown in the figure as red boxes. The website has since been replaced, but an archived version is available via the Internet Archive Wayback Machine at (*Timeline* 2018). However, according to the website for health data in Norway (*Variabler* 2023), the current number of participants is 172,472. A possible explanation is that the current dataset excludes some of the initially recruited participants. Nevertheless, the difference is insignificant for the paper's content and conclusions.

Contributions by the author of this thesis

- The purpose and scope of the research were formulated collaboratively with the paper's co-author.
- Reviewed literature and gathered information on systems epidemiology and NOWAC through meetings with researchers in the field
- Developed and conceptualized the suggested systems architecture
- Wrote the original draft of the manuscript and made the figures

/ 15

Paper II

Autostrata: Improved Automatic Stratification for Coarsened Exact Matching

The second paper presents a novel stratification algorithm for use with Coarsened Exact Matching (CEM) and is relevant to observational studies in epidemiology, such as studies using hybrid designs. For example, case-control studies nested within prospective cohorts involve selecting cases and controls that will be part of the analysis. The purpose is to match comparable cases and controls or balance a sample to control for confounder bias, which is part of the study design as a step before data analysis. The method is a contribution to the pre-analysis step of the data analysis pipeline described in Section 9.3.

It should be noted that the stratification algorithm can be used independently for other purposes and is generalizable.

Contributions by the author of this thesis

- Identified the research gap and developed the methodology
- Made the implementations and conducted the experiments
- Wrote the original draft of the manuscript and made the figures

/16

Paper III

Greedy Knot Selection Algorithm for Restricted Cubic Spline Regression

The contribution of the third paper fits into the statistical analysis and modeling step of the data analysis pipeline described in described in Section 9.3. It presents a knot selection process for restricted cubic spline (RCS) regression. RCS regression is frequently used in epidemiology, and other fields, to model non-linear relationships between predictor and response variables.

Contributions by the author of this thesis

- Suggested and developed the novel method for knot selection
- Made the implementations and conducted the experiments
- Wrote the original draft of the manuscript and made the figures



Paper IV

Schema-Based Priming of Large Language Model for Data Object Validation Compliance

The paper provides an empirical basis for using natural language for defining analysis pipelines, which can be used as part of the proposed platform's no-code or low-code UI. The paper includes a user scenario example where a scientist designs an analysis pipeline for air pollution data. Using an example relating to a field different than epidemiology was a deliberate choice to indicate the potential generalizability of the thesis' proposed platform outside epidemiology.

The developments around LLMs and the features offered by OpenAI's GPT-4 models are racing at the time of writing. The paper was submitted 21st of April, 2023. Almost two months later, 14th of June, OpenAI announced the *function calling* feature for GPT-4 (<https://platform.openai.com/docs/guides/gpt/function-calling>), which uses a system message prompt to prime the model to learn the signature of functions by a schema-like definition. This enables the model to automatically output a JSON structure representing the function call in the response text as a response to user prompts. GPT-4 does not call the function, but an application integrating it can extract the JSON representation of the function call and use it. The novel approach presented in *Schema-Based Priming of Large Language Model for Data Object Validation Compliance* has many similarities. However, the preprint was publicly available

before OpenAI announced their function calling API. Further, the intended use and proposed solution presented in the paper are more advanced and extensive than the function calling feature in GPT-4, for example, building a validated structured data object over multiple prompts and responses.

The paper is under revision, but the original manuscript is included due to its relevance to the thesis and the similarities with the method used later by the mentioned OpenAI API.

Contributions by the author of this thesis

- Found the research gap, and the aim and scope of the research were decided collaboratively with the co-author
- Suggested and developed the method
- Made the implementations and conducted the experiments
- Wrote the original draft of the manuscript and made the figures

/18

Concluding Remarks

A case has now been made for a platform providing a collaboration saas targeting groups of researchers using data from a common cohort for a multitude of studies having hybrid designs. It has also proposed an architecture and design for the unique part of the system. The main lesson learned is that code generation can open a door otherwise shut when providing a public cloud solution to facilitate users working with regulated data available exclusively in a secured, non-public environment, especially when users cannot freely install software not provided by default. Another lesson learned is that the great utility of systems for designing studies and data analyses may sometimes be found in the collaborative aspects and what the users as a group contribute to the solution.

The theme of the thesis also serves as the overall context within which the four included papers belong. All the papers are on topics related to analytical observational studies and analysis pipelines for studies utilizing a common data source as a basis for hybrid designs. Paper I establishes the original setting from which the thesis has evolved. Papers II and III and the report on smoking-variable wrangling contribute to the analysis pipeline's wrangling, pre-analysis, and analysis steps. Paper IV is related to no-code and model-driven development but employs a large language model (LLM) as part of a natural-language user interface (NLUI) for designing pipelines. In future work, the NLUI could be combined with a visual design environment for no-code development. Further, instead of requiring the researcher to type the text manually, a modern *automatic speech recognition* (ASR) technology could be

used to transcribe speech to text. OpenAI's Whisper is an example of a recent ASR system (Radford, J. W. Kim, et al. 2023).

The cross-disciplinary focus has been at the intersection of computer science, statistical methods, and epidemiology, and particularly on analytical observational studies and hybrid designs. Here, it is important to realize that other research fields also conduct observational studies. For example, Paper IV deliberately uses an example related to air pollution instead of epidemiology. At the same time, the principles are equivalent to the pipeline definitions and runnable pipelines described in the thesis. From the thesis perspective, the reason was to illustrate that the proposed principles for designing studies and data pipelines are generalizable beyond epidemiology—however, this is particularly true if the data must be kept in a secured IT environment with little room for installing extra software.

18.1 Limitations

Through the Ph.D. project, the author has explored the field of analytical observational studies within epidemiology from a cross-disciplinary point of view, examining limiting factors and uncovering opportunities. In the process, it has been important to see the overall picture, as well as to dive deep into the matter to understand the issues better.

However, a Ph.D. project has limited scope due to limited time and resources. Implementing and deploying a complete solution is left to future work and requires a project with more resources and people involved. The author strongly believes that a platform providing a collaborative SAAS, as described in the thesis, can be of great value when it comes to the design of new studies and helping to foster a collaborative culture within research groups. At the same time, there are many opportunities for computer science student projects to develop various parts of the platform and introduce new features. The author also envisages possible other studies regarding the platform, such as user studies.

Further developing the proposed platform can be interesting and fruitful for epidemiology and computer science researchers. Therefore, the author hopes the thesis can be used as the starting point of a larger project than a single Ph.D. project can cover.

Part V

Included Papers

Paper I

The Beauty of Complex Designs

Jo Inge Arnes and Lars Ailo Bongo

Advancing Systems Epidemiology in Cancer. Scandinavian University Press (Universitetsforlaget) (2020). DOI: 10.18261/9788215041193-2020-0200

License: CC BY 4.0



2. The Beauty of Complex Designs

Jo Inge Arnes and Lars Ailo Bongo

Abstract The increasing use of omics data in epidemiology enables many novel study designs, but also introduces challenges for data analysis. We describe the possibilities for systems epidemiological designs in the Norwegian Women and Cancer (NOWAC) study and show how the complexity of NOWAC enables many beautiful new study designs. We discuss the challenges of implementing designs and analyzing data. Finally, we propose a systems architecture for swift design and exploration of epidemiological studies.

Keywords Systems epidemiology | Norwegian Women and Cancer | study designs | hypothesis exploration | computer systems

INTRODUCTION

Analytical observational epidemiology was, and primarily still is, about disease risk estimation. In the past, most studies used simple case-control designs with data from questionnaires, registers, and health records. The analyses relied on Cox and classical survival analysis methods. Because case-control designs are prone to selection and recall bias, prospective cohorts with nested designs are increasingly used, but typically still focus on risk estimation. However, there is a shift in epidemiology towards more basic research in which we study how diseases affect biological systems at a biomolecular level over time – for example, to understand the dynamics of human carcinogenesis.

This shift was motivated by the sequencing of the human genome, officially completed in April 2003 (The Human Genome Project), which led to the incorporation of genetic variants into epidemiological studies, primarily single nucleotide polymorphisms (SNPs). SNPs are ideal as exposures because they do not change over a lifetime. Hence, risks can be estimated using classical statistical methods. There are also many hospital and research biobanks with samples usable for SNP

analyses, such as biobanks incorporated in the European Prospective Investigation into Cancer and Nutrition (EPIC) (Bingham and Riboli 2004). In the ensuing decade, considerable resources were spent on genome-wide association studies (GWAS), but the studies repeatedly failed to find robust, replicable associations between SNPs and common diseases (Lund and Dumeaux 2008). The focus, therefore, shifted to functional genomics to find biological markers associated with environmental exposures, lifestyle, age, or disease.

In 2008, Lund and Dumeaux (Lund and Dumeaux 2008) introduced systems epidemiology and proposed the globolomic design. Systems epidemiology incorporates functional genomics and observes how diseases affect human biological systems over time. The globolomic design extends the existing prospective design by integrating functional genomics analyses from blood and tissue. In 2015, Lund, with collaborators, introduced a processual approach to systems epidemiology (Lund et al. 2015). The processual approach differs from traditional risk-related research in that we view disease as a multi-stage process and use functional genomics to observe disease-associated changes over time. In connection with the new direction in epidemiology, there was a need for new statistical methods. An example is a statistical method for longitudinal gene expression analysis using the concept of curve groups (Lund et al. 2016, Chapter 8), developed in cooperation with the Norwegian Computing Center.

Omics (Vailati-Riboni et al. 2017) plays an essential part in systems epidemiology. The different omics are, unlike genes, affected by exposures and diseases. By integrating omics in nested case-control studies, we can find altered levels of gene expressions or methylation that are biological markers of the disease. For example, studies have discovered changes in pre-diagnostic DNA methylation associated with breast and lung cancer risk (Baglietto et al. 2017, Fasanelli et al. 2015, van Veldhoven et al. 2015). Other studies have found changes in the inflammatory transcriptome in adults related to early-life socioeconomic status (Castagne et al. 2016). We can also use other types of biological data that contain changes associated with a disease, including epigenetics, gene expressions, proteins, and metabolites. Finally, we can combine different types of omics and observe them together in a multi-omics approach (Hasin et al. 2017).

In systems epidemiology, we observe how diseases affect human biological systems at the molecular level over time in order to gain more knowledge about the mechanisms involved throughout the natural history of a disease. The development of cancer, for example, is a multi-stage process (Foulds L 1958, Grizzi and Chiriva-Internati 2006). The omics may be affected differently at different stages of the process. Thus, the temporal aspects are essential – for example, the time to

diagnosis. Systems epidemiology can help to bridge the gap between epidemiology and research in biological sciences. The study findings can provide input into research on molecular level biological systems, which can enhance our understanding of diseases, e.g. through pathway analysis (Garca-Campos et al. 2015). We can, therefore, see systems epidemiology as a shift in epidemiology from applied research towards basic research. The emphasis on the dynamic nature of biological systems and processes in systems epidemiology can be seen as a counterpart to systems biology, which is a discipline that seeks to determine how complex biological systems function by integrating experimentally derived information through mathematical and computing solutions (Institute of Systems and Synthetic Biology).

We can integrate systems epidemiological designs into existing prospective studies if the studies include omics and relevant questionnaire data. The Norwegian Women and Cancer study is an example of a complex prospective study with extensive data from questionnaires and registers, nested studies, different types of preserved biological samples, and omics data.

However, many opportunities remain unexplored due to the time-consuming and expensive steps required to conduct a full systems epidemiological project. We could reduce the problem by making it possible to quickly design studies and explore potential hypotheses at an early stage, before starting thorough research projects.

In this paper, we show that many novel systems epidemiological studies are possible by utilizing existing data from population-based prospective cohort studies. We also propose a computer systems architecture enabling the swift design of studies and exploration of hypotheses.

COMPLEX DESIGNS

Systems epidemiological study designs can be nested within existing cohort studies, such as the Norwegian Women and Cancer (NOWAC) study. The novel studies thus become part of a larger, complex design. Here, we describe the NOWAC study and data types, and we show that the existing cohort enables many novel study design possibilities. We give a stepwise example of a systems epidemiological design process. We also provide examples of two other variations of study designs to show that there are several ways to design studies. Lastly, in this section we discuss the potential for realizing more of the potential for designing studies and exploring hypotheses.

NORWEGIAN WOMEN AND CANCER STUDY

In this paper, we use the Norwegian Women and Cancer (NOWAC) Study (Lund et al. 2008) to describe the systems epidemiological design process. NOWAC is a population-based prospective cohort study approved by the Regional Committee for Medical Research Ethics and the Norwegian Data Inspectorate (PREK NORD 141/2008 Biobanken KVINNER OG KREFT). It was initially designed for breast cancer research and has later been used to research other types of cancer. The cohort includes 172 556 Norwegian women born between 1926–1965 (Gram et al. 2013). Invitations to the study were sent by mail in different batches for different time periods (The Norwegian Women and Cancer Study, NOWAC). Most of the women were recruited between 1991–1997 (179 387 invited, 102 540 recruited) and 2003–2006 (130 577 invited, 63 232 recruited) (Lund et al. 2008). All of the invited women had been randomly drawn from the Norwegian Central Person Register. Each woman in the study has participated in surveys with questionnaires covering a wide range of topics, from smoking, alcohol, diet, and physical activity to the use of oral contraceptives and hormonal replacement therapy, reproductive history, and diseases in the family.

The women have answered follow-up surveys with intervals of between four to six years, resulting in a total of one to four answered questionnaires per woman. The latest follow-up was in 2017. NOWAC periodically updates data with information from the Norwegian Cancer Registry and the Cause of Death Registry.

There are also blood and tissue samples. The number of women in NOWAC born 1943–1957 is about one-third of all Norwegian women born in those years, and between 2003–2006, the NOWAC postgenome cohort study (Dumeaux et al. 2008) collected blood samples from about 50 000 of these participants. At the time of blood sampling, the participants filled out an accompanying two-page questionnaire. The samples were collected using the PAXgene™ Blood RNA System (PreAnalytiX GmbH, CH-8634 Hombrechtikon, Switzerland) with buffers specially designed for the conservation of RNA (Barnung et al. 2018).

Other types of samples also exist for a smaller portion of the women, such as biopsies from both malignant tumors (Dumeaux V 2017) and healthy tissue (Chapter 4). NOWAC produced its first microarray-based gene expression dataset in 2009 and later miRNA, DNA methylation, metabolomics, and RNA-Seq datasets (Fjukstad 2019).

The samples have been preserved with the future in mind. Assessment of the mRNA quality in whole blood samples after 15 years has been reassuring (data not shown). We are still early in the post-genomic era, and the omics field is rapidly evolving. In the future, new or improved types of assays will be developed. We can

then use the preserved samples together with these assays. Also, tissue and blood samples can be analyzed in new ways as new areas of interest emerge in cancer research. For example, the immune system's role in cancer is promising (de Visser et al. 2006). In the future, other areas may attract attention.

Systems epidemiology's use of biological samples from human participants has a number of advantages compared to the alternatives. In biomedical research, for example, it is common to conduct experiments either on live laboratory animals (in vivo) or in Petri dishes and test tubes (in vitro). It is reasonable to assume that there are relevant differences between humans and laboratory mice that must be taken into account when studying human diseases (Breschi et al. 2017, Mestas and Hughes 2004). In their daily lives, humans experience very different exposures compared to laboratory mice. Systems epidemiological designs make it possible to investigate gene expression profiles resulting from the complex real-life situations of the participants, with hundreds of different exposures that interact with genetic predispositions to cancer (Lund and Dumeaux 2008).

A prospective study, such as NOWAC, will often start as a cross-sectional study in which data collection is done at a defined time. The study will usually involve surveys about the past and data originally collected for other purposes. Cross-sections of the cohort can be made, but the temporality desired in a prospective study is still missing. For each following year, some percentage of the participants will be affected by cancer or another disease, which forms the basis for the prospective aspect of the study. Additionally, the cohort needs to be followed up. Follow-ups of a cohort can involve mailing follow-up questionnaires, updating data from disease and cause-of-death registers, and possibly blood and tissue sampling.

The NOWAC study was designed as a prospective cohort study from the beginning. The aim of the study was initially to research hormonal contraceptives and breast cancer risk, but the surveys included questions covering a far broader scope. This is the reason why NOWAC can be used to research many other cancers and risk factors. In addition to the original study, there are different nested studies within NOWAC. These are mostly case-control studies. An advantage of nesting case-control studies in prospective cohorts is the reduction of recall and selection bias. Other study designs can be nested, as well. Some studies exist that only use the controls from a nested case-control study.

We can use the data in NOWAC for many novel epidemiological studies (Figure 2.1). Before any diagnosis, most participants have answered multiple surveys and donated blood samples. Data from the surveys give an insight into the participants' prior exposures and risk factors related to lifestyle, family history, socioeconomic status, and health status. This information is supplemented with data from passive

follow-up based on cancer and death register data, and active follow-up based on collaboration with 11 major Norwegian hospitals and the Norwegian Breast Cancer Group (NBCG). Blood samples were collected and stored in a way that makes new functional genomics analyses possible. Because the blood was collected before diagnosis, the time between blood sampling and diagnosis varies for different cases. In addition to the pre-diagnostic blood samples, some post-diagnostic samples were collected as well. NOWAC also includes tissue samples from hospital biobanks for many of the participants that developed cancer. The study even has four hundred biopsies from healthy women; see Chapter 4. The blood and tissue samples are analyzed using several omics technologies. All this data can be combined in many different ways, enabling many system epidemiology studies, which we will show in the following section.

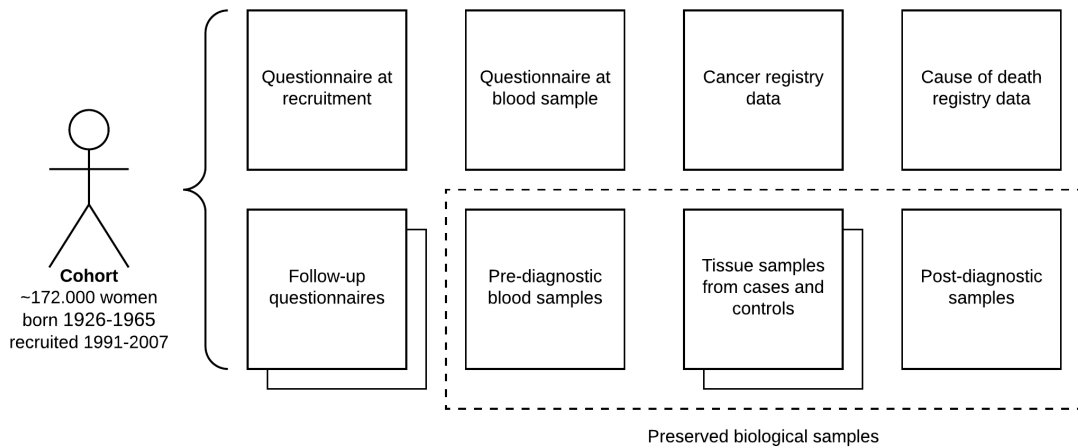


Figure 2.1. NOWAC cohort overview; biological samples and data types.

DESIGNING SYSTEMS EPIDEMIOLOGICAL STUDIES

Here, we describe how novel systems epidemiological studies can be designed using data from NOWAC. We first describe limitations of the data material before moving on to the many possible combinations of data that exist. We then provide an example of the design process.

Limitations

Before we describe the many possibilities in a prospective cohort, we first discuss the limitations. One type of limitation is when the data material does not contain

the necessary information. A trivial example is that a cohort without male participants probably does not have the data needed for prostate cancer research.

When it comes to questionnaire data, it is important to be aware that not all groups respond to surveys to the same extent. The validity of studies concerning high alcohol consumption can be problematic because people who suffer from alcoholism answer questionnaires to a lesser extent than others. Consequently, data on this group may be insufficient. However, studies involving other groups can still be valid. The validity of the questionnaire items can also be of concern—have the participants understood the questions? Furthermore, the types of data obtainable from samples are limited by the technology used for collection and cold storage. To conserve RNA in blood, we must use technologies such as PAXgene or similar.

The size of the cohort is another limiting factor. In studies involving subgroups, statistical power can often become a problem due to too few participants. One way of counteracting the problem is through international collaborations. The European Prospective Investigation into Cancer and Nutrition (EPIC) (Bingham and Riboli 2004) is one such international collaboration. EPIC is one of the largest prospective cohort studies in the world. It has 521 000 participants and has been followed for almost fifteen years. The cohort is composed of other cohorts from ten European countries, including NOWAC.

A significant problem internationally is the follow-up of mortality and disease. In Norway and the other Nordic countries, follow-up is easier thanks to public register data. All Nordic countries have a central person register, cause-of-death register, disease registers, and other public registers. Although not perfect in every respect, the Nordic registers have long been celebrated as a ‘gold mine’ for research (van der Wel et al. 2019).

The many possible studies

When we design a study, there are many types of choices that we can make depending on the research hypothesis. The different types of choices comprise a high number of possible studies when combined.

Figure 2.2 shows the intersection of seven different types of choices as separate dimensions. There are many options for each dimension, and the intersection of the dimensions results in an ample decision space where each combination is a potential study design. In the following, we describe the different choice dimensions.

The first dimension (1) concerns choices related to the study design's time aspect, which is an integral part of most epidemiological study designs. In system epidemiological designs, we define a timeline dimension explicitly. We can divide the timeline into the time before diagnosis, time of diagnosis, or time after diagnosis. For some samples, such as biopsies taken at diagnosis, the time will coincide with the time of diagnosis, but we can combine this with other samples taken before or after diagnosis. We can also further divide the timeline into intervals, e.g. 0–1 years before diagnosis, 2–3 years before diagnosis, and 3–5 years before diagnosis, which is useful for statistical analyses.

The second dimension (2) is the exposures and risks dimension. Many different types of exposures can increase the risk of a condition. In NOWAC's prospective questionnaires, we find information about each participant's risk factors, such as lifestyle, use of medication, conditions in the family, number of births, and much more. Additionally, genetic variants can be viewed as risk factors that can be identified by analyzing blood samples.

The third dimension (3) is the different types of measurements and assays that we can choose. In the NOWAC context, each assay is an omics or multi-omics assay – for example methylation, gene expressions, and metabolomics.

However, there are more than three dimensions. Instead of adding more axes, we label the remaining dimensions with lower case letters a–d on a cube (see label 4 in the figure). Each cube in the figure will have these four additional choice dimensions, which differentiate the many possible studies.

The fourth dimension (4a) represents the possible diagnoses that can be studied. In NOWAC, we have information about various diagnoses from the Norwegian Cancer Registry and the Cause of Death Registry.

The fifth dimension (4b) is the participant selection dimension. This dimension concerns the criteria for choosing and grouping participants for the study. A typical example is a case-control study in which we select cases from the cohort based on criteria that we choose. We then choose controls nested in the cohort matched on the cases. The criteria that we use to match controls to cases can vary from study to study, while selecting controls with the same sex and similar age since the case is quite common. There will usually be far more possible controls than cases available for selection in a study. A ratio of about a thousand to one is not uncommon. The statistical power is dependent on the number of available cases and the number of controls drawn for each case.

The sixth dimension (4c) is the sample type dimension. Usually, it matters where the analyzed sample was acquired from; it can be a blood sample, a tissue sample, or a sample of specific types of immune cells. We can compare results from differ-

ent sample types from each participant, such as comparing methylation levels in peripheral blood and tumor tissue.

The seventh dimension (4d) applies to stratification and de-confounding. The purpose is to adjust for underlying factors that skew the results, and we usually use exposure and risk factor data for this. An example of how we can adjust for smoking exposure when analyzing biomarkers for lung cancer is given in a later description of a three-level study design.

We have now described the many available choices that exist when designing studies. Each dimension consists of many options, and the number of possible studies becomes very large when we combine different dimensions. The reason for the high number of combinations is that the number of options for each dimension must be multiplied together. The total number of combinations then becomes: *(The number of ways to arrange the timeline) * (The number of exposures) * (The number of available measurements and assays, e.g. for single or multi-omics) * (The number of available diagnoses) * (The number of ways to select participants) * (All sample types and relevant combinations) * (The de-confounding and stratification factors)*

After we have chosen the study parameters from the described dimensions, we will have a clearer understanding of the selection of data we need for a study. The next step is to apply the data selection to systems epidemiological designs.

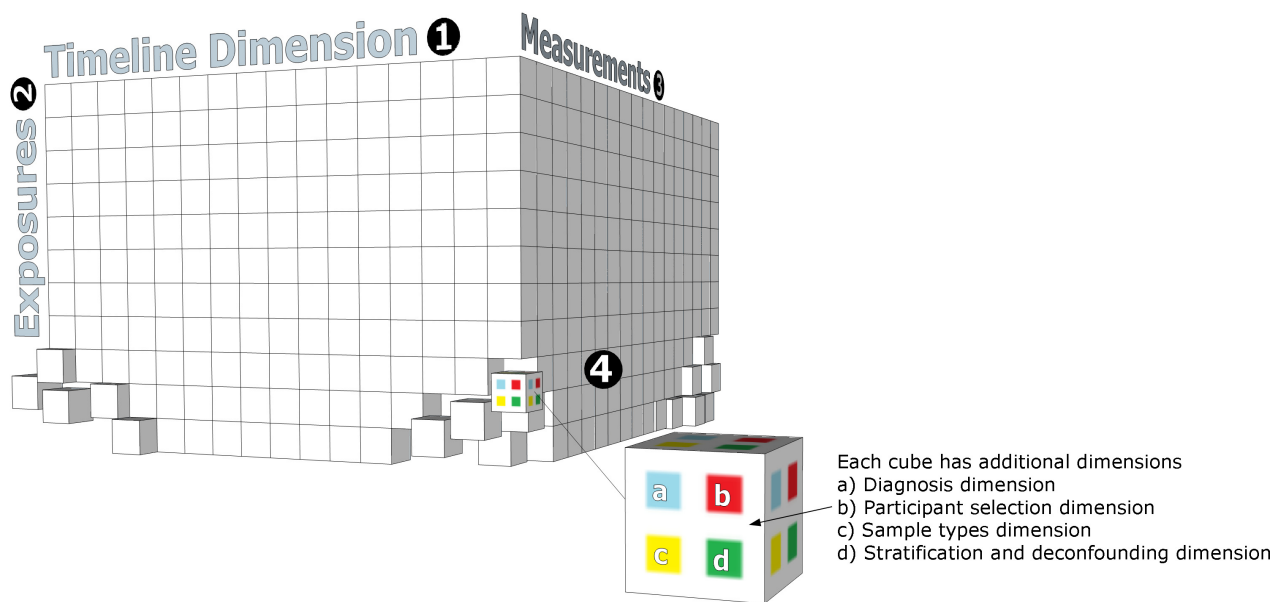


Figure 2.2. The different dimensions that can be combined for each study design.

Applying data to systems epidemiological designs

After deciding on the parameters and data for our study, we apply the data within a systems epidemiological design. We now give a stepwise example of a systems epidemiological design process using existing data from a prospective cohort study with omics data, such as NOWAC.

In systems epidemiology, imagine that we organize our sample data points along several axes, where one is the timeline (Figure 2.3). We usually split the timeline into the time before diagnosis, of diagnosis, and after diagnosis. It is also possible to split the timeline by an event other than the diagnosis. The decision on how to split the timeline was described earlier as one of the dimensions from which we choose our study parameters.

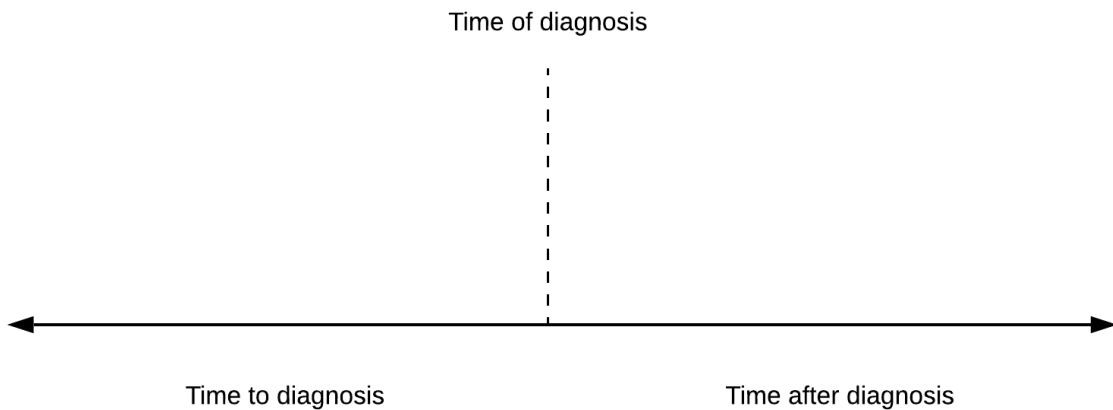


Figure 2.3. Time to diagnosis, time of diagnosis, and time after diagnosis.

Each sample in our data has a temporal distance to the time of diagnosis (Figure 2.4). We therefore place the data points on the timeline relative to how long before or after diagnosis the sample was collected. The second axis is a value axis. The values of the data points can be the raw measured values, such as the expression levels for a gene, but they are often the results of a function that takes one or more measured values as parameters. For example, the vertical position of the data point may represent the difference between cases and controls (Formula 2.1).

$$f(x_{case}, x_{ctrl}) = \log_2(x_{case}) - \log_2(x_{ctrl})$$

Formula 2.1. In the formula, x is a case-control pair's expression levels for a gene or other omics value.

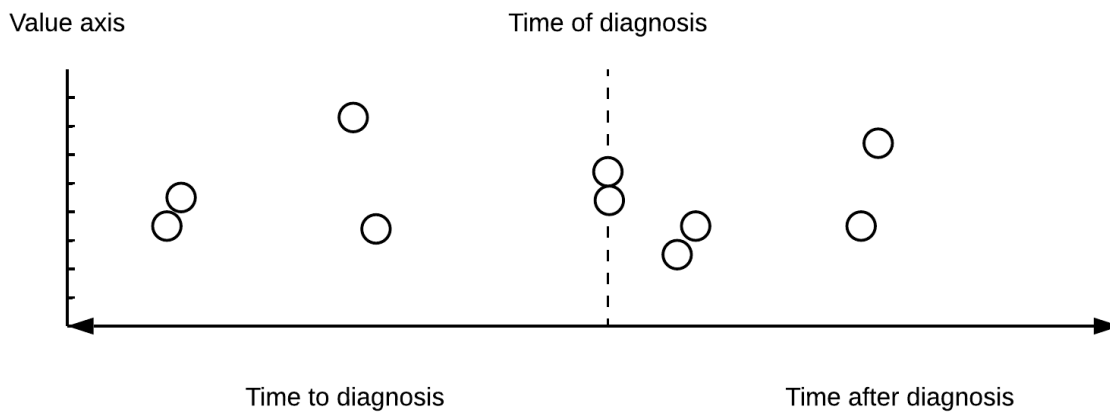


Figure 2.4. Sample data points positioned by distance from diagnosis. The value axis does not have to be linear; it can be logarithmic or other.

Next, we can group data points into strata that we are interested in comparing (Figure 2.5). By observing data points at a group level, we can envision a curve or trajectory for each stratum (Figure 2.6). If we compare the trajectories and find significant differences between the strata, this could potentially be of importance not only for future research on differential diagnosis or prognosis, but also for understanding which biological systems are involved.

It is not mandatory to stratify by grouping data points as described. If the data point values come from a function that represents a comparison of different samples, then this too is a type of stratification. When using Formula 1 for data point values, the height of the curve is a case-control comparison. Consequently, multiple levels of stratification can be achieved through a combination of grouping and use of functions.

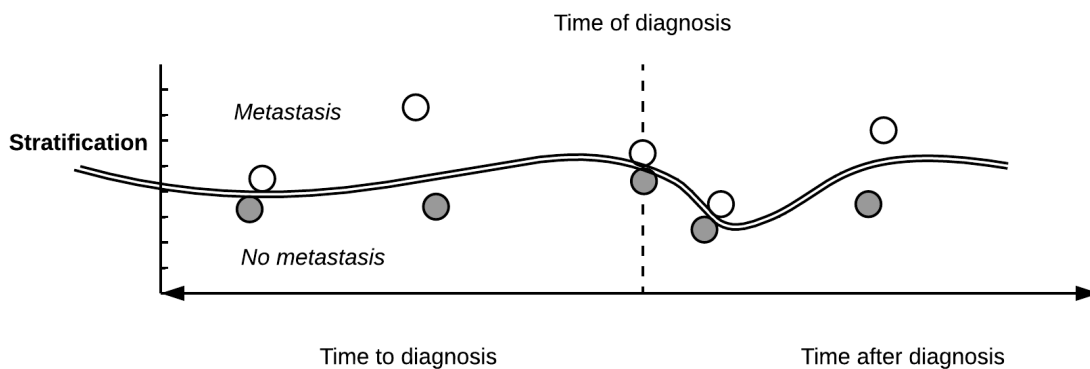


Figure 2.5. Stratification of data points. In this example, the white-filled circles represent women with metastasis, and the grey-filled circles represent women without.

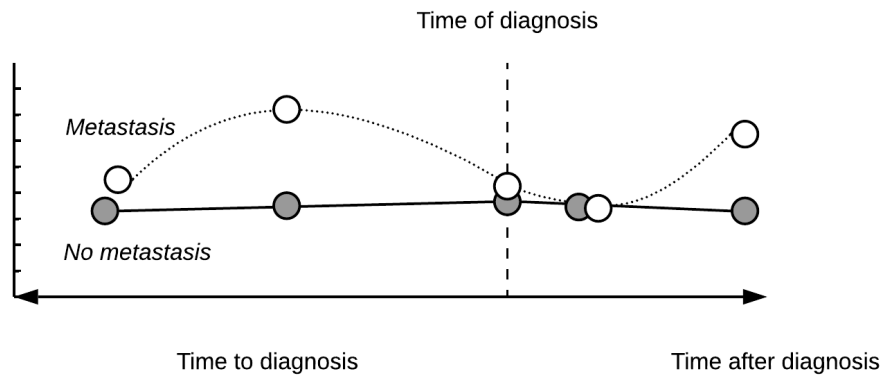


Figure 2.6. An illustration of estimated curves or trajectories for each stratum. The curves for the two strata are different.

Because the measured values are from biological processes that interact as part of a system, it is interesting to compare the curves of many types of values simultaneously (Figure 2.7). The figure shows three curves per stratum, one for each type of gene expression.

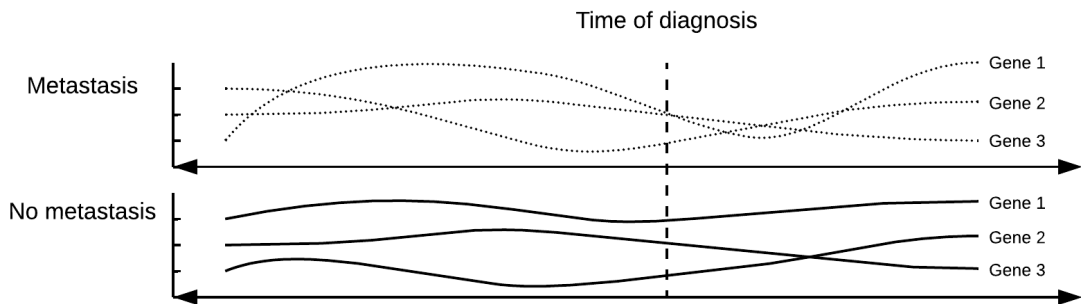


Figure 2.7. For each biological sample, we can measure the levels of many different expressed genes. For each, we can imagine a separate curve per strata. In the illustration, only the expression levels for “Gene 1” differ between the two strata. Note that we are not restricted to gene expressions. Other omics can be used.

However, the reality is more challenging than illustrated in Figure 2.7. For example, we can measure the expression levels for 19 950 protein-coding genes from each blood sample and present each expressed gene as a separate curve along the timeline. Curves for other omics can be included as well, such as methylation. The results can thus consist of thousands of intersecting curves per stratum, which is too much information to be presented as an overview of the data. Therefore, we must use other techniques for analyzing the data. Many methods exist for analyzing high-dimensional omics data. Usually we use methods related to clustering or dimensionality reduction techniques for high-dimensional data (Breschi et al.

2017). Examples of dimensionality reduction techniques include principal component analysis (PCA), multidimensional scaling (MDS), and t-distributed stochastic neighbor embedding (tSNE). An alternative approach is to map the omics data to a biological context, e.g. we can map gene expressions to where they occur in biological pathways. We are also interested in including the temporal aspect as part of the data analysis, which is a hallmark of systems epidemiology.

We have now described how studies can be designed by applying existing cohort data, for example, a combination of questionnaire data and high-dimensional molecular data from NOWAC. The steps in the design process described in this section can be summarized as:

- Establish an axis for the time to diagnosis (or another event) and an axis for values
- Define strata
 - For example, cases with spread or without spread
- Calculate data point values and position them in the coordinate system
 - The basis for the values is analyzed samples, taken from different participants at different times. Pre-diagnostic samples acquired from the cases will usually have different distances to the time of diagnosis
 - The data point values can be the raw measured values from samples, but more often we use derived values from computations and statistical methods that include values from case-control pairs
- Imagine curves for each similar type of data point belonging to the same stratum
 - For example, all data points for a specific mRNA that involve cases with spread belong to the same curve
- For high-dimensional data, there will be too many curves to comprehend, and advanced clustering or dimensionality reduction techniques are thus needed
- Compare the strata to find differences
 - Statistical methods, data explorations, and visualizations

TWO ALTERNATIVE TYPES OF STUDY DESIGN

In the previous section we based the studies on comparing cases and controls, but there are other possibilities. Here we describe two design variations.

The NOWAC study has tissue samples that we can analyze and compare to peripheral blood. That is, we compare samples from different locations in the same person instead of between cases and controls. NOWAC includes case-control pairs for which diagnostic blood and tissue samples exist both for cases and matching

controls, which means that women allowed health-care professionals to take biopsies of healthy tissue for research purposes. For these participants, we can design studies that compare tissue and blood samples and also include the case-control aspect (Dumeaux et al. 2017).

Figure 2.8 It is also possible to define study designs with more than one level of nesting. For example, we can create a three-level design comprised of the cohort, a nested case-control study, and a cross-sectional study that only includes the controls (Figure 2.8). The following case exemplifies this type of design: For some diseases, such as lung cancer, a large percent of the cases has a history of smoking exposure. As a result, it can be hard to separate the early biological effects of cancer from the effects of smoking. We can solve this problem by first finding biomarkers for smoking exposure in the controls. In the cross-sectional study, the controls are stratified based on exposure data from the cohort’s prospective questionnaires. The gene expressions are then analyzed to find the biological markers of smoking. In the parent case-control study, the findings can be used for de-confounding purposes to prevent smoking markers from being misinterpreted as cancer markers. A study similar to this has been conducted by (Baiju et al. 2020) as part of the Id-Lung project. The same type of design was used by to demonstrate altered gene expression levels in the NOWAC cohort associated with coffee consumption (Bar-nung et al. 2018).

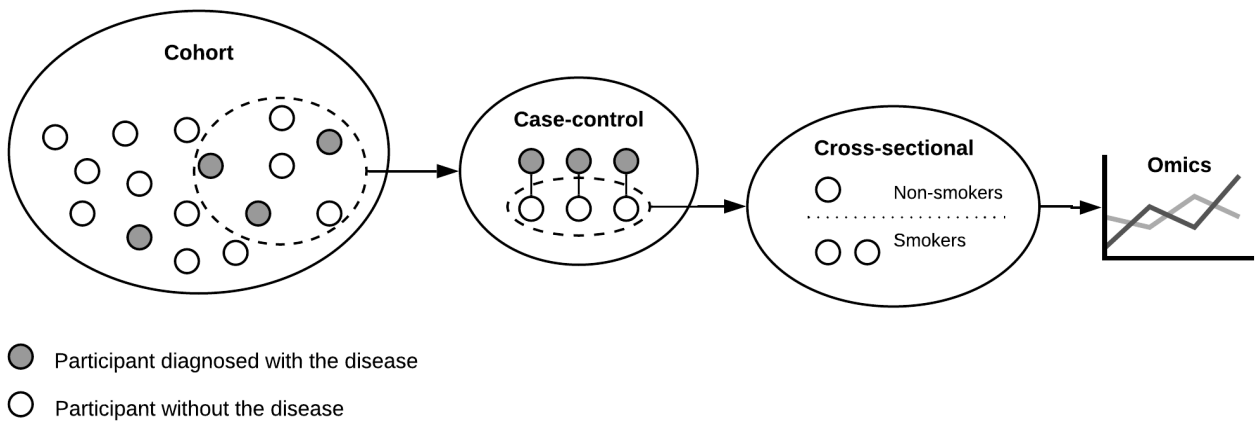


Figure 2.8. An illustration of a three-level design. Case-control pairs are selected from the prospective cohort. The cross-sectional study selects controls from the case-control study. The controls are stratified by exposure, which in this case is smoking status. The smoking statuses are calculated from the cohort study’s questionnaires, and the biological samples are also from the cohort. The gene expression data is part of the case-control study. The cross-sectional study analyzes the gene expressions to find exposure markers.

TOWARDS REALIZING THE POTENTIAL

We have shown that it is possible to combine data in numerous ways to design many different studies. Unfortunately, a lot of time and resources are needed to carry out full epidemiological studies. Consequently, many opportunities that lie in the prospective cohorts may be left unrealized.

If, instead, we had carried out lightweight studies in a simple way in advance wherein we could quickly explore potential hypotheses, then we could have had a better starting point when deciding whether it would be worth going ahead with larger projects.

To realize more of the potential that lies in the NOWAC data and similar studies, we suggest that a computer system should be created that supports the rapid design of studies, analysis of data, and exploration of hypotheses. In the following sections, we propose a computer systems architecture for this purpose.

COMPUTER SYSTEMS ARCHITECTURE

In systems epidemiology, we design complex studies with many types of data, including high-dimensional molecular data. Computer systems are essential for managing data and performing computations. In the previous section we discussed the possibility of a computer system helping to realize more of the potential in cohort data by enabling the users to explore different hypotheses quickly. However, no such unified system presently exists for systems epidemiology.

Here, we propose a systems architecture that enables the swift design of studies, analysis of data, and exploration of hypotheses. The aim is to explore different hypotheses quickly at a preliminary stage of research, or explained with a metaphor: “We wish to explore the data by swimming and delving into it.” (Lund 2019, personal communication)

There exists a range of software tools and systems that are used in systems epidemiology. Examples are tools that are concerned with processing omics data in pipelines, data management, or reproducibility in science. Fjukstad et al. 2018 (Chapter 3) used a combination of such tools to organize data storage and documentation and to standardize the analysis of gene expression data in NOWAC. Various unrelated tools and scripts for statistical analyses of omics also exist. None of these tools and systems constitute a unified system for the swift design of studies, epidemiological analysis, and exploration of hypotheses. We present a high-level, conceptual architecture for this missing system.

Figure 2.9 shows a conceptual view of the proposed system’s architecture. The system is illustrated as having a pipelined architecture in which one part’s output becomes the next part’s input. The arrows between the parts represent the flow of data. Each part may be composed of loosely coupled subsystems.

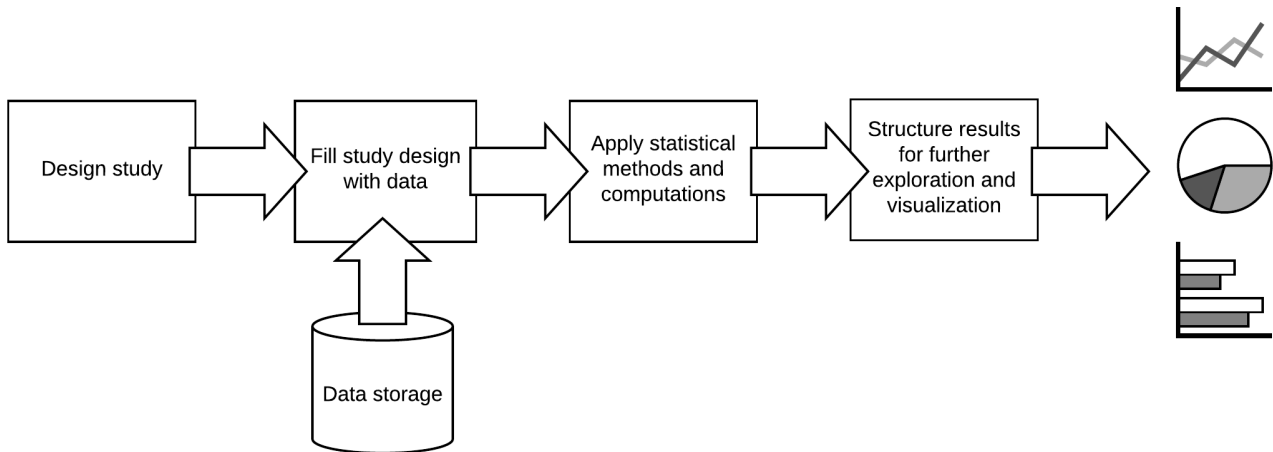


Figure 2.9. A high-level conceptual view of a computer system for systems epidemiology.

In addition to designing each part of the system, we must design good abstractions for the interfaces between them. We can view most of the system’s parts as separate black boxes; the outside does not know the details of how the part functions on the inside. The outside can only interact with it through limited interfaces and is not permitted to manipulate its inner state and workings directly. An abstraction is a well-defined view or model that only includes what is relevant and excludes all that is irrelevant. The art is to define abstractions that are correct for use, flexible and general enough to include relevant variations, yet simple and coherent. We commonly prefer interfaces and data structures with these properties. We implement them by using the available features for declaring data types, functions, and schemas in our programming languages, software frameworks, and environments. The conscious use of abstractions when designing systems is an important tool for avoiding accidental complexity, and it provides the system with clean and simple-to-understand façades (Kleppman 2017). Abstractions also help to clearly separate the system’s different concerns and make it more flexible to changes.

First, we provide an example use case describing the system from the researcher’s point of view. Next, we discuss the five main parts of the system. We additionally touch upon the importance of reproducibility in science.

Example use case: Design a study in an interactive notebook

In this section, we describe how the researcher can use the system through an interactive notebook. Interactive notebooks are increasingly popular in data science and scientific computing. The notebooks enable researchers to create dynamic documents containing a mix of text and runnable code fragments. We use the notebooks as interactive development environments and share them with others. Two examples of notebook environments are R Notebook (Chapter 3.2 in Xie et al. 2019) and (The Jupyter Notebook). We provide a casual use case (Cockburn 2000) describing a notebook approach to designing studies.

A researcher wants to design a study in order to explore a hypothesis. The researcher has already opened a notebook and loaded the required packages belonging to the system. The researcher types in and runs a simple command (or function-call) telling the system to create a workspace for the study. The system creates a data structure representing an empty workspace, which becomes available in the researcher's notebook. Included in the workspace is a default study design specification. The researcher specifies the study's overall design by adding groups and stratifications to the design specification. The system keeps a data structure representing this design within the study design specification. The researcher specifies the data sets that will be used, including the target versions. The system keeps this information in the workspace. The researcher then defines queries for the different groups and strata. The system keeps the queries but does not yet run them to fetch data. At this point, the researcher wants to inspect the data, which is an optional step. The system runs the queries on demand and makes the data available. After inspecting the data, the researcher defines how data will be analyzed by composing statistical methods and computations from standard or custom packages. These can be associated with specific groups or strata, and sequences of computations can be defined. The system keeps this in the workspace. The researcher instructs the system to execute the entire study, and the system executes the study by fetching necessary data and running computations as specified. It does this by delegating work to the storage and computational systems, such as data lakes and Apache Spark. It makes the resulting data available in the researcher's notebook environment. The researcher can then further explore and visualize the results with other tools.

The researcher can save the workspace at any point. Previously saved workspaces can be loaded and run. The researcher can modify individual parts of the workspace and execute the updated study.

Design study

To easily specify new study designs, we must provide a user interface (UI) to the system that is user-friendly and practical. Several options exist:

- A graphical UI for specifying study designs
- A human-readable text-based format for defining studies (XML, JSON, YAML)
- A software package integrated into a development environment commonly used in the researcher's field (R-studio)
- A domain-specific language (DSL) for defining study designs

Regardless of how we present the study design specification UI to the researcher, the specified designs must internally be represented in a machine interpretable manner that is useable later for the automatic execution of the study. The study design specifications describe what the researcher wishes to do, but not the details of how. The exact decision on how data retrieval and execution is performed is left to other parts of the system. This type of abstraction ensures that changes in implementation details, or even the replacement of whole subsystems, can be contained to the parts that retrieve data and execute the study without requiring changes to other parts. Equally important, the abstraction makes it possible automatically to optimize how the study is performed.

Data storage

Data is central in epidemiological research, but managing all the technical aspects of data is complicated and bears little relevance to the researcher's aims. For example, a considerable amount of time is spent on data wrangling due to impractical data structures or lack of consistent structures. Each project typically operates on smaller, custom data sets that have been extracted manually from the primary data sets. The data sets are stored in simple text-based formats on shared disks. The included fields and names are inconsistent across data sets. Sometimes the researchers will make personal copies of the data set file, with various changes that they have made. With the advent of multi-omics, the amount of data can potentially become very large, which will require a more professional approach to data management. The system should hide the technical details surrounding data and instead provide the researchers with simple, uniform data access.

Today, a variety of production-quality data storage solutions are available. It is crucial to investigate which type of solution best suits the system because there are significant differences between them. Examples of storage types are:

- Relational database management systems (RDBMS), including data warehouses: PostgreSQL, MS-SQL
- Key-value stores: Redis, Memcached
- Column stores or column formats: Cassandra, Parquet
- Graph databases: Neo4j, OrientDB
- Files in distributed file systems: Hadoop Distributed File System (HDFS), Tachyon
- A combination of the above, termed polyglot persistence (Sadalage and Fowler 2013)
- Data lakes (Miloslavskaya and Tolstoy 2016): Azure Data Lake, AWS Data Lake

A layer of abstraction should be created for easy and uniform access to the data, hiding the underlying data structures and storage systems. By abstracting the underlying storage mechanisms away from the rest of the system, it is easier to evolve or replace the storage solution as we discover opportunities for improvements. ADAM (Massie et al. 2013) is a set of formats, APIs, and processing stage implementations for genomic data. It has a layered design with a “narrow waist” in the middle, also termed an hourglass model (Beck 2019). The narrow-waist layer consists of a data schema, implemented with Apache Avro (The Apache Avro Project) that separates the details of the storage layers from the overlying layers. A similar approach may prove useful in our system.

Fill study design with data

After specifying a study design, the researcher must be able to query and retrieve the data for the study. First, one or more data sources are chosen. We should enable access to the data in a uniform manner and structure the data according to standard schemas. Next, the researcher defines queries that select and transform data for the study’s different groups and strata, such as cases, controls, with spread, without spread. The queries are attached to the study design specification.

From the technical side, the queries should be attached to the study design but not immediately executed. The system should be allowed to run queries in the same context as the computations. This can prevent inefficient spilling of data to disk between the steps. It can also enable automatic query optimizations. There are several options for query languages, e.g., the query syntax could be SQL-like or fluent (Fowler 2005). LINQ (Torgersen 2007) or Resilient Distributed Datasets (RDD) (Zaharia et al. 2012) are examples that support deferred execution and both types of syntaxes.

The resulting data must have a structure recognizable by the computational and statistical methods in the next step of the workflow. Again, we need good abstractions.

Computations and statistical methods

The researcher should be able to choose from ready-made calculations and statistical methods and possibly define custom ones. Functions for common computations and statistical methods can be packaged in a reusable manner that is independent of a particular study. The statistical methods for curve groups (Lund E 2016) and classify strata (Holden 2015) are candidates for such packages. Novel statistical methods for systems epidemiology will likely be developed in the future. The system must support both ready-made packages, as well as custom packages. A statistician can implement functions, possibly in collaboration with scientific programmers, and epidemiologists can then apply the functions in various studies. A challenge is to define standards for functions and packaging that covers the needs of existing and future statistical methods.

The computations involved in omics analysis are often time-consuming and resource-heavy. Care should be taken to choose an underlying platform that performs well for the computations encountered in systems epidemiology. Apache Spark (Zaharia et al. 2010) is a unified analytics engine for large-scale data processing that could be used as an integral part of the system. Recent versions of Spark support R (The R Project for Statistical Computing), which is a programming language and environment for statistical computing often used in epidemiology.

Structure results for further exploration and visualization

After applying computations and statistical methods, it should be easy for the researcher to explore and visualize the data further. Because many general-purpose tools and software packages already exist that are excellent for data exploration and visualization, the results generated by the system should be usable within the context of such software packages and tools. We can achieve this by structuring data in a standard format so that the researcher can either use the result datasets directly or import them into their software tool of choice, such as an R environment.

Reproducibility

It has been claimed that there is a reproducibility crisis in science. *Nature* (Baker 2016) asked 1576 researchers questions about reproducibility. They found that 90% answered that there was either a slight or significant crisis. More than 70% had tried and failed to reproduce other scientists' experiments. More than half of the scientists had experienced that they were unable to reproduce their own exper-

iments. There are several reasons for the crisis – for example, selective reporting or low statistical significance. At other times it can be challenging to know how to repeat the experiment correctly. In the latter case, we can benefit from having a system that can automatically rerun previous experiments using the same steps and data.

The system's study design specifications, dataset selections, queries, and statistical methods can be saved together as a complete workflow. As long as the underlying data stay unchanged, the experiments can be reloaded and automatically repeated. The system must track changes to data and support data versioning. By specifying target data versions for the workflows, we can ensure that the experiment's data stays the same between runs.

CONCLUSION

We have described the complex NOWAC study, the many different types of data, and that the data can be combined in a large number of ways. The many combinations allow us to create many new system epidemiological study designs. We have also given a step-by-step example of a system epidemiological design.

The beauty of complex studies such as NOWAC is the opportunities for new studies that arise. However, opportunities can be lost because extensive studies are time-consuming and costly. By finding a quick way to create designs using existing data, we can perform initial explorations to investigate if a hypothesis is worth researching more extensively.

As a solution, we have proposed a computer systems architecture to support the swift design of system epidemiological studies and exploration of hypotheses.

ACKNOWLEDGEMENTS

We wish to thank Professor Eiliv Lund for providing invaluable input and feedback on the paper.

REFERENCES

- Apache Spark [Internet]. Available from: <https://spark.apache.org>
- Baglietto L, Ponzi E, Haycock P, Hodge A, Bianca Assumma M, Jung CH et al. DNA methylation changes measured in pre-diagnostic peripheral blood samples are associated with smoking and lung cancer risk. *Int J Cancer*. 2017 Jan 1; 140(1): 50–61. Available from: <https://onlinelibrary.wiley.com/doi/full/10.1002/ijc.30431>
- Baiju N, Sandanger TM, Sætrum P, Nøst TH. Gene expression in whole-blood reflects smoking exposure among cancer-free women in the Norwegian Women and Cancer postgenome cohort. Submitted.
- Baker M. 1,500 scientists lift the lid on reproducibility. *Nature*. 2016 May 26; 533(7604): 452–454. Available from: <https://www.nature.com/articles/533452a>
- Barnung RB, Nøst TH, Ulven SM, Skeie G, Olsen KS. Coffee Consumption and Whole-Blood Gene Expression in the Norwegian Women and Cancer Post-Genome Cohort. *Nutrients*. 2018 Aug 9; 10(8): 1047.
- Beck M. On The Hourglass Model. *Communications of the ACM*. 2019 Jul; 62(7): 48–57. Available from: <https://cacm.acm.org/magazines/2019/7/237714-on-the-hourglass-model/fulltext>
- Bingham S, Riboli E. Diet and cancer--the European Prospective Investigation into Cancer and Nutrition. *Nat Rev Cancer*. 2004 Mar; 4(3): 206–215. Available from: <https://www.nature.com/articles/nrc1298>
- Breschi A, Gingeras TR, Guigo R. Comparative transcriptomics in human and mouse. *Nat Rev Genet*. 2017 Jul; 18(7): 425–440. Available from: <https://www.nature.com/articles/nrg.2017.19>
- Castagne R, Kelly-Irving M, Campanella G, Guida F, Krogh V, Palli D, et al. Biological marks of early-life socioeconomic experience is detected in the adult inflammatory transcriptome. *Sci Rep*. 2016 Dec 9; 6: 38705. Available from: <https://www.nature.com/articles/srep38705>
- Cockburn A. Writing effective use cases. Series: The Crystal Collection for software professionals. 1st ed. Addison-Wesley Professional; 2000. pp 304.
- Dumeaux V, Børresen-Dale AL, Frantzen JO, Kumle M, Kristensen VN, Lund E. Gene expression analyses in breast cancer epidemiology: the Norwegian Women and Cancer postgenome cohort study. *Breast Cancer Res*. 2008 Feb; 10(1): 1–8. Available from: <http://breast-cancer-research.com/content/10/1/R13>
- Dumeaux V, Fjukstad B, Fjosne HE, Frantzen JO, Holmen MM, Rodegerdts E, et al. Interactions between the tumor and the blood systemic response of breast cancer patients. *PLoS Comput Biol*. 2017 Mar 7; 13(9): e1005680. Available from: <https://doi.org/10.1371/journal.pcbi.1005680>
- de Visser KE, Eichten A, Coussens LM. Paradoxical roles of the immune system during cancer development. *Nat Rev Cancer*. 2006 Jan; 6(1): 24–37. Available from: <https://www.nature.com/articles/nrc1782>
- Fasanelli F, Baglietto L, Ponzi E, Guida F, Campanella G, Johansson M, et al. Hypomethylation of smoking-related genes is associated with future lung cancer in four prospective cohorts. *Nat Commun*. 2015 Dec 15; 6: 10192. Available from: <https://www.nature.com/articles/ncomms10192>
- Fjukstad B. Toward Reproducible Analysis and Exploration of High-Throughput Biological Datasets [Doctoral thesis]. Tromsø: UiT, The Arctic University of Norway; 2019. 149 pp. Available from: <https://munin.uit.no/handle/10037/14576>

- Fjukstad B, Shvetsov N, Nøst TH, Bøvelstad H, Halbach T, Holsbø E et al. Reproducible data management and analysis using R. bioRxiv. 644625, in press. Available from: <https://www.biorxiv.org/content/10.1101/644625v1>
- Foulds L. The natural history of cancer. *J Chronic Dis*. 1958 Jul; 8(1): 2–37. Available from: <https://www.sciencedirect.com/journal/journal-of-chronic-diseases/vol/8/issue/1>
- Fowler M. FluentInterface. At martinfowler.com [Internet]. Accessed 06.06.2019. Available from: <https://martinfowler.com/bliki/FluentInterface.html>
- Garcia-Campos MA, Espinal-Enriquez J, Hernandez-Lemus E. Pathway Analysis: State of the Art. *Front Physiol*. 2015 Dec 17; 6: 383. Available from: <https://www.frontiersin.org/articles/10.3389/fphys.2015.00383/full>
- Gram IT, Sandin S, Braaten T, Lund E, Weiderpass E. The hazards of death by smoking in middle-aged women. *Eur J Epidemiol*. 2013 Sep 29; 28(10), 799–806. Available from: <https://link.springer.com/article/10.1007/s10654-013-9851-6>
- Grizzi F, Chiriva-Internati M. Cancer: looking for simplicity and finding complexity. *Cancer Cell Int*. 2006 Feb 15; 6(1): 4. Available from: <https://cancer-ci.biomedcentral.com/articles/10.1186/1475-2867-6-4>
- Hasin Y, Seldin M, Lusis M. Multi-omics approaches to disease. *Genome Biol*. 2017 May 5; 18(1): 83. Available from: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1215-1>
- Holden L. Classify strata. Oslo: Norwegian Computing Center; SAMBA/11/15; 2015. pp 28. Available from: https://www.nr.no/directdownload/1426685952/classify_strata_holden2015.pdf
- Imperial College London. Institute of Systems and Synthetic Biology [Internet]. Accessed: 06.06.2019. Available from: <https://www.imperial.ac.uk/systems-biology/about-the-institute/>
- Kleppman M. Designing Data-Intensive Applications: the big ideas behind reliable, scalable, and maintainable systems. 1st ed. Sebastapol, CA: O'Reilly Media; 2017. pp 569. Available from: <https://books.google.no/books?id=zFheDgAAQBAJ&lpg=PP1&lr&hl=no&pg=PP1#v=onepage&q&f=false>
- Lund E. Personal communication. Meeting at Institute for Informatics about BoCD. Tromsø, 2019.
- Lund E, Dumeaux V. Systems epidemiology in cancer. *Cancer Epidemiol Biomarkers Prev*. 2008 Nov; 17(11): 2954–2957. Available from: <https://cebp.aacrjournals.org/content/17/11/2954.long>
- Lund E, Dumeaux V, Braaten T, Hjartaker A, Engeset D, Skeie G et al. Cohort profile: The Norwegian Women and Cancer Study--NOWAC--Kvinner og kreft. *Int J Epidemiol*. 2008 Feb; 37(1): 36–41. Available from: <https://academic.oup.com/ije/article/37/1/36/763947>
- Lund E, Holden L, Bøvelstad H, Plancade S, Mode N, Günther CC, et al. A new statistical method for curve group analysis of longitudinal gene expression data illustrated for breast cancer in the NOWAC postgenome cohort as a proof of principle. *BMC Med Res Methodol*. 2016 Mar 5; 16(1): 28. Available from: <https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-016-0129-z>
- Lund E, Plancade S, Nuel G, Bøvelstad H, Thalabard JC A processual model for functional analyses of carcinogenesis in the prospective cohort design. *Med Hypotheses*. 2015 Oct; 85(4): 494–497. Available from: <https://www.sciencedirect.com/science/article/pii/S0306987715002704?via%3Dihub>

- Massie M, Nothaft FA, Hartl C, Kozanitis C, Schumacher A, Joseph AD et al. ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing. Technical Report No. UCB/EECS-2013-207. Electrical Engineering and Computer Sciences, University of California at Berkeley; 2013. pp 22. Available from: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-207.pdf>
- Mestas J, Hughes CC. Of mice and not men: differences between mouse and human immunology. *J Immunol*. 2004 Mar 1; 172(5): 2731–2738. Available from: <https://www.jimmunol.org/content/172/5/2731.long>
- Miloslavskaya N, Tolstoy A. Big Data, Fast Data and Data Lake Concepts. *Procedia Computer Science*. 2016; 88: 300–305. Available from: <https://reader.elsevier.com/reader/sd/pii/S1877050916316957>
- Kvinner og kreft, Blodprøve og biopsi [Internet]. Accessed: 28.07.2020. Available from: <https://site.uit.no/kvinnerogkreft/blodprove-og-biopsi/>
- National Institute of Health, National Human Genome Research Institute. The Human Genome Project [Internet]. Accessed: 15.11.2019. Available from: <https://www.genome.gov/human-genome-project>
- Norwegian Computing Central [Internet]. Available from <https://www.nr.no/en>
- Notebook. Chapter 3.2 in Xie Y, Allaire JJ, Grolemond G (eds) *R Markdown: The Definitive Guide* [Internet]. Accessed: 15.11.2019. Available from: <https://bookdown.org/yihui/rmarkdown/notebook.html>
- Sadalage PJ, Fowler M. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. New Jersey: Pearson Education, Inc; 2013. pp 164.
- The Apache Avro Project [Internet]. Available from: <https://avro.apache.org>
- The EPIC Study [Internet]. Available from: <https://epic.iarc.fr>
- The Jupyter Notebook [Internet]. Available from: <https://jupyter.org>
- The Norwegian Women and Cancer Study, NOWAC [Internet]. Accessed: 06.06.2019. Available from: <https://site.uit.no/nowac/methodological-description/timeline/>
- The R Project for Statistical Computing [Internet]. Available from: <https://www.r-project.org>
- Torgersen M. Querying in C#: how language integrated query (LINQ) works. In: *Proceeding OOPSLA '07 Companion to the 22nd ACM SIGPLAN conference on Object-oriented programming systems and applications companion*, Montreal, Quebec, Canada, Oct 21–25, 2007. New York: ACM Press; 2007. pp 852–853.
- UiT The Arctic University of Norway. Id-Lung [Internet]. Available from: https://en.uit.no/forskning/forskningsgrupper/gruppe?p_document_id=507532.
- Vailati-Riboni M, Palombo V, Looor JJ. What Are Omics Sciences? In: Ametaj B (eds) *Periparturient Diseases of Dairy Cows*. Cham: Springer; 2017. pp. 1–7. Available from: https://link.springer.com/chapter/10.1007%2F978-3-319-43033-1_1#citeas
- van der Wel KA, Östergren O, Lundberg O, Korhonen K, Martikainen P, Andersen AN, Urhoj SK. A gold mine, but still no Klondike: Nordic register data in health inequalities research. *Scand J Public Health*. 2019 Aug;47(6):618–630. Available from: <https://journals.sagepub.com/doi/10.1177/1403494819858046>
- van Veldhoven K, Polidoro S, Baglietto L, Severi G, Sacerdote C, Panico S, et al. Epigenome-wide association study reveals decreased average methylation levels years before breast cancer diag-

nosis. Clin Epigenetics. 2015 Aug 4; 7: 67. Available from: <https://clinicalepigeneticsjournal.biomedcentral.com/articles/10.1186/s13148-015-0104-2>

Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, San Jose, CA, USA, Apr 25–27, 2012. Berkeley: USENIX Association Berkeley; 2012(2–2). Available from: <https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf>

Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. In: HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, Boston, MA, UAS, Jun 22–25, 2010. Berkeley: USENIX Association Berkeley; 2010(10–10): p. 95. Available from: https://www.usenix.org/legacy/events/hotcloud10/tech/full_papers/Zaharia.pdf

Paper II

Autostrata: Improved Automatic Stratification for Coarsened Exact Matching

Jo Inge Arnes, Alexander Hapfelmeier, and Alexander Horsch

In *Proceedings of the 18th Scandinavian Conference on Health Informatics*. Linköping Electronic Conference Proceedings, LiU Electronic Press (2022). DOI: 10.3384/ecp187030

License: CC BY 4.0

Autostrata

Improved Automatic Stratification for Coarsened Exact Matching

Jo Inge Arnes¹, Alexander Hapfelmeier², and Alexander Horsch³

¹UiT The Arctic University of Norway, Department of Computer Science, Tromsø, Norway, jo.i.arnes@uit.no

²Technical University of Munich, Institute of AI and Informatics in Medicine, München, Germany

³UiT The Arctic University of Norway, Department of Computer Science, Tromsø, Norway

Abstract

We commonly adjust for confounding factors in analytical observational epidemiology to reduce biases that distort the results. Stratification and matching are standard methods for reducing confounder bias. Coarsened exact matching (CEM) is a recent method using stratification to coarsen variables into categorical variables to enable exact matching of exposed and nonexposed subjects. CEM's standard approach to stratifying variables is histogram binning. However, histogram binning creates strata of uniform widths and does not distinguish between exposed and nonexposed. We present Autostrata, a novel algorithmic approach to stratification producing improved results in CEM and providing more control to the researcher.

Keywords

Analytic epidemiology, confounder bias, stratification, coarsened exact matching, algorithms

1 INTRODUCTION

Epidemiologists conduct analytical observational studies [1] to investigate associations between exposures and outcomes. Instead of assigning a treatment or exposure to the participants of a randomized experiment [2], we rely on observations of the subjects in their usual environment with minimal interference. There are many established ways of designing observational studies, from cross-sectional, cohort, and case-control studies to more complex prospective cohorts with several nested case-control and cross-sectional designs [3, 4, 5]. A common theme for these is awareness of biases. Confounding factors [6, 7, 8] are a common source of bias that can, if measured, be adjusted for in the analysis [9, p. 1020]. Stratification [10], for example, can control for confounding by dividing study subjects into groups based on observed confounders. Iacus et al. [11] present the *coarsened exact matching* (CEM) method that adjusts for bias by turning confounder covariates into categorical variables through stratification, which we can then use to match comparable subjects exactly. Blackwell et al. [12] introduce a Stata (<https://www.stata.com>) implementation of CEM, and Iacus et al. [13] provide an implementation for R (<https://www.r-project.org>). In addition, a web page with an overview of implementations for other programming languages and platforms is available (<https://gking.harvard.edu/cem>). The same webpage also informs that CEM is officially qualified for scientific use by the U.S. Food and Drug Administration. The CEM implementations let users create strata manually or use automatic stratification. The built-in automatic stratification creates uniform width bins by applying general rules of thumb for constructing histograms. The three binning algorithms included in both Stata and R are Sturges' rule [14], Scott's rule [15], and Freedman-Diaconis' rule [16]. Additionally, Stata includes an implementation of Shimazaki-Shinomoto's rule [17].

Blackwell et al. [12, p. 534] demonstrate that manually defining strata based on domain knowledge can sometimes give better results than the current automatic approach. In their example, the manually defined strata are less imbalanced while giving a higher number of matched units. However, according to King et al. [18, p. 439], researcher biases are highly likely to affect qualitative choices even when researchers attempt to avoid them. 'The literature makes clear that the way to avoid these biases is to remove researcher discretion as much as possible,' following King. On the other hand, the general histogram binning rules do not support the specific challenges of stratifying confounders:

- The histogram binning algorithms do not distinguish between different groups of units and include no concept of matching.
- They do not take into account multivariate imbalance between groups.
- The strata have uniform widths, i.e., all strata for a covariate have the same width.
- The researcher cannot in advance give parameters to influence the stratification process.

Against this background, we researched and developed a novel algorithmic approach to the stratification problem that addresses the shortcomings above. We implemented the algorithm and experimentally compared it to CEM's built-in histogram binning with good results.

We conclude the introduction with a brief example of Autostrata's applicability to health-related studies. For instance, say we want to study if coffee consumption is associated with a beneficial effect on the risk of liver cancer. In the respective observational study, we must be cautious of possible systematic differences between the compared groups, such as smoking habits. Failing to adjust for these differences can challenge the validity of the results. Autostrata improves such adjustments when using

The 18th Scandinavian Conference on Health Informatics, Tromsø, Norway, August 22-24, 2022. Organized by UiT The Arctic University of Norway. Conference Proceedings published by Linköping University Electronic Press at <https://doi.org/10.3384/ecp187>. © The Author(s). This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/>

CEM. The method creates more precise results and keeps more study participants included in the analysis.

After the introduction, the structure of the paper is as follows: First, we provide essential background for understanding the problem. Next, we describe our approach and algorithm. We then present the experiments and results, followed by a discussion. Last, we briefly touch upon related work before concluding.

1.1 A note on terminology

The paper primarily uses the terms *treated and controls* instead of *exposed and nonexposed* due to their use in CEM and the general causal inference literature. In addition, although case-control studies are different from cohort studies that focus on exposed and unexposed, they are interchangeable in this paper because we concentrate on the stratification of confounder covariates in isolation from these differences. Further, we use the general statistical term units instead of subjects, individuals, or study participants often seen in epidemiology.

2 BACKGROUND

Before presenting our approach, we provide the background necessary to understand the challenges of the stratification problem.

2.1 Confounding

We often assess whether the risk of a health event (outcome) is increased or decreased among an exposed or treated group compared to a control group. To quantify the relationship between an exposure or treatment and the outcome, we calculate risk ratios, odds ratios, or other measures. However, other factors not directly under investigation can skew the results or even lead us to the opposite conclusion of what is correct. Figure 1 illustrates how confounding factors influence both the exposure and the outcome. Note that the confounder is not in the direct causal pathway between the two. Also, a relevant property of confounders is that the compared groups have differently distributed values for the confounder covariate. If the confounders are measured and included in the dataset, we can adjust for confounders during analysis, which is the purpose of the stratification discussed in this paper. It is worth noting that according to Wacholder et al. [9, p. 1020], the use of stratification or matching can, in effect, adjust for unknown or unmeasured confounders through reduced variability because this variability is measured conditionally on the levels of other studied variables.

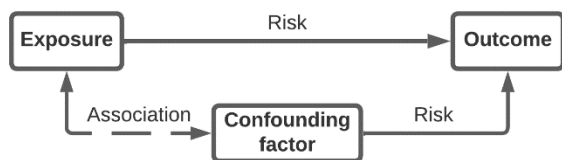


Figure 1 shows an exposure that is associated with a risk of an outcome. The confounding factor is associated with both the exposure and the outcome without being in the direct causal pathway of the two.

2.2 Counterfactuals and imbalance

The Neyman-Rubin causal model (RCM) [19] is one of the notable influences on the understanding of causal inference in observational studies. According to the model, to estimate the effect of a treatment on an outcome, we should

ideally compare the treated subjects with the same subjects without treatment. Except for the treatment, all other conditions must be the same, including the time. The latter is a counterfactual and is impossible to observe. We instead compare to relatively similar, untreated controls. However, the treated and controls in our sample are often systematically different or imbalanced for the confounding factors, which leads to bias. Lowering this imbalance between treated and controls to make them more similar is thus a strategy to reduce the bias.

2.3 Coarsened exact matching

As earlier explained, the confounder covariates are distributed differently for the compared groups. Thus, we can view the bias as stemming from an imbalance in the data. Coarsened exact matching (CEM) [11] is a method for adjusting confounder bias as a preprocessing step before analysis. It belongs to a class of monotonic imbalance bounding (MIB) methods, enabling the researcher to set a maximum imbalance between treated and controls for the confounder covariates or reduce the maximum imbalance for a covariate independently of others. The theoretical foundation of CEM is outside the scope of this paper, but its use is relatively straightforward.

We partition the confounder covariates into subintervals. Each subinterval then represents a single value of a categorical variable. For example, a covariate for years of education can be partitioned into subintervals representing the highest level of education instead. In CEM, this is called coarsening and opens for simple, exact matching of similar treated and control units. It additionally helps balance the sample by pruning treated and control units without suitable matches. The coarsening is temporary and not passed to subsequent analysis steps.

The described coarsening corresponds to stratification. We stratify each covariate, and each treated and control unit will then belong to a multi-dimensional stratum. Although the current CEM software packages use uniform width histogram binning for automatic stratification, CEM as a method is not restricted to strata of uniform widths. For example, manual stratification and non-uniform widths are supported. Autostrata is an alternative approach to automatically stratifying covariates, which constructs strata of non-uniform widths.

2.4 Imbalance and unmatched trade-off

The most commonly described imbalance measure for CEM involves the relative difference between the number of treated and control units per stratum. However, the software packages use an imbalance measure based on a per stratum difference in means between the covariate values for the two groups as default. This is similar to what Appendix B of [11, p. 34] describes. We thus base our approach on the latter.

As shown in Figure 2, two strata with the same number of treated and control units can have a different internal imbalance because the covariate means are different for the groups. Nevertheless, the maximum imbalance is bounded by the stratum widths because the differences cannot be greater than the widths. Therefore, the narrower the stratum is, the lower its maximum imbalance. The lowest maximum imbalance is when each stratum only has a single unit or equal-valued units. A stratum with only one type of unit contributes zero to the imbalance, while multiple equal-valued units have an imbalance of zero. The challenge is that there is a trade-off.

CEM prunes unmatched units from the sample. If all units in a stratum are from the same group, these units are unmatched and discarded. Recall that the confounder covariates for treated and control units have different distributions. Hence, various degrees of overlap and densities will be found along the covariate axes, restricting how narrow a stratum containing both types of units can be. As we decrease the maximum imbalance, the number of unmatched units generally increases, and vice versa. Autostrata aims to lower this trade-off.

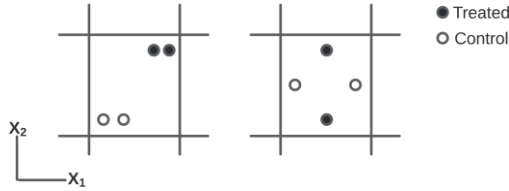


Figure 2 illustrates two strata for covariates x_1 and x_2 . Both strata have two treated and two controls, but the left stratum has a higher mean difference. Also, the maximum difference is bounded by the width between the stratum edges.

2.5 Stratification problem properties

Before concluding the background section, we describe a few properties of the stratification problem relevant to solving it algorithmically.

First, the number of relevant stratum edges is finite. The reason is that a stratum edge for a covariate can be placed anywhere between two adjacent observations without changing stratum memberships. If an observation coincides with an edge, it belongs to the higher stratum. The exact position of an edge does not matter, only that it separates two adjacent observations for the given covariate. Neither do multiple stratum edges between two neighboring observations change any memberships. Further, if two or more observations have equal values for a covariate, they cannot be separated by adding stratum edges for the given covariate. Conclusively, the maximum number of relevant stratum edges equals the number of distinct values per covariate.

Second, the number of possible combinations of the stratum edges, from including no edge to including all edges, grows exponentially with the number of distinct covariate values, i.e., the problem space is non-polynomial.

Figure 3 shows all possible combinations of stratum edges for four distinct values, organized as a tree of nodes. The number of new stratifications that can be made by adding one stratum edge to a given stratification is illustrated in Figure 4.

We can deduce the number of different stratifications possible for a covariate. Let S be the set of possible stratifications for a covariate with n distinct values. Then the cardinality, $|S|$, is:

$$|S| = 1 + (n - 1) + \sum_{i=2}^{n-1} 2^{i-2}(n - i) = 2^{n-1}$$

Given m covariates, the total number of combinations, $|S_{tot}|$, becomes:

$$|S_{tot}| = \prod_{i=1}^m |S_i| = \prod_{i=1}^m 2^{n_i-1}$$

For cases where all $n_i = n$ are equal:

$$|S_{tot}| = |S|^m = 2^{m(n-1)}$$

Thus, the *state space* of the problem grows exponentially with increasing numbers of distinct values and covariates. Furthermore, considering that each stratification can contain relatively many multi-dimensional strata and that we must compute imbalance measures and the number of unmatched units for each stratification, it quickly becomes computationally infeasible to perform a brute-force search through all combinations to find an optimal solution with the resources typically available to researchers.

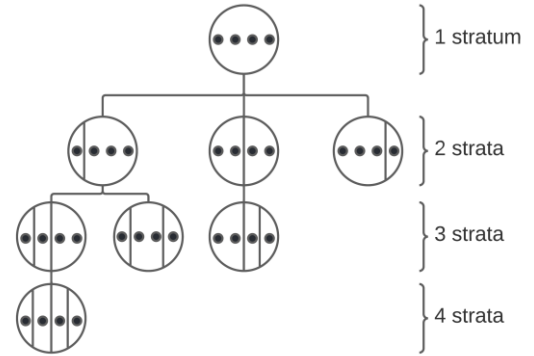


Figure 3. All possible stratifications of a covariate with four distinct observed values. The four values are illustrated as black dots within the tree nodes, and the stratum edges as vertical lines between the dots.

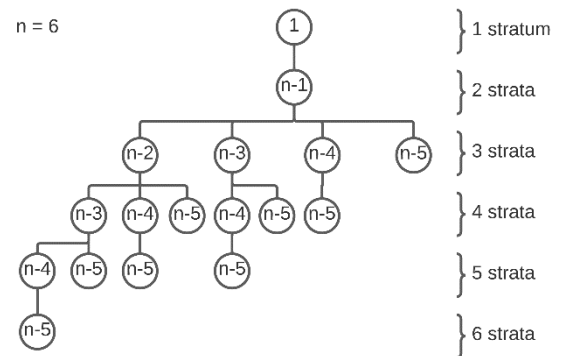


Figure 4. This tree illustrates a pattern in the number of different stratifications that can be made as we move from a given parent to a child node by adding a new stratum edge, as in Figure 3. In this case, the number of distinct values is $n=6$.

3 AUTOSTRATA

We now present Autostrata, a novel algorithmic approach for improved stratification of confounder covariates for CEM. Improving CEM's standard stratification method—histogram binning—is not trivial. However, analysis results need to be as free of bias as possible to avoid them from being invalid. Often, the imbalance is higher than we wanted, the number of unmatched units is high, or both. Autostrata aims to lower the trade-off between the imbalance and the number of unmatched units. Figure 5 shows a comparison of histogram binning and Autostrata.

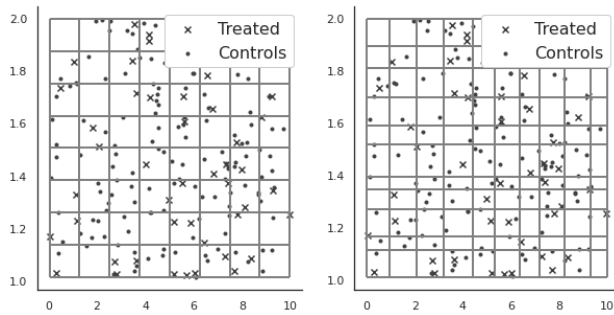


Figure 5 shows two stratifications for the same two-covariate dataset. The left plot is from histogram binning, and the right is from Autostrata. Each grid cell is a two-dimensional stratum. On the left, the strata have uniform widths. On the right, the strata widths are non-uniform.

3.1 Overall approach

This section gives an overall description of the Autostrata approach and explains its reasoning.

The generic histogram binning rules used in CEM work surprisingly well for stratification in our context. Therefore, understanding the underlying reasons is invaluable to improving the results: Any stratum containing both treated and controls is valid. Also, the sample’s total maximum imbalance will be lower if the strata are narrower. To construct strata spanning over a mixture of treated and control units, regions of common support must be present for the sample, i.e., there must be some overlap in the distributions for treated and controls. The treated and controls in regions with sparse or no overlap are further apart and more dissimilar than units in denser and more overlapping regions. Because we usually have a reasonable common support level, the uniform width strata will readily contain both treated and control units. Further, units in the sparser and less overlapping regions are more likely to be pruned, as they should. These factors contribute to why histogram binning works well. Conclusively, knowing these factors makes it reasonable to assume that much of the potential for improvement is in the regions where the distributions for treated and controls overlap most.

Autostrata’s strategy is to construct narrow strata while keeping the number of unmatched units low. The strata can be of varying widths. Having narrower strata on average is equivalent to more strata. We thus start with an initial stratification state where all possible stratum edges for all covariates are included (see section 2.5). This state represents the narrowest stratification that is relevant. All units will be in a stratum containing only a single unit or same-valued units. From there, we iteratively remove one edge at a time. This edge can belong to any of the covariate dimensions.

In its simplest form, the algorithm does not consider widths but removes edges one by one until the number of unmatched units is as low as requested by an input parameter. The main selection criterion for removing an edge, per iteration step, is the edge that gives the most significant reduction in unmatched units when removed. Removing a stratum edge for one dimension (covariate) merges one or more strata divided by stratum edges for other dimensions. Merging strata for a given covariate results in strata that are wider, so the increase in the average width of the strata for a covariate is strictly monotonic.

The crux of the algorithm is: For each stratum edge that we remove from the initial state, the average maximum

imbalance increases. If the algorithm reaches the requested maximum number of unmatched in fewer steps, i.e., by removing fewer edges, the *average maximum imbalance* will be lower than if more steps are spent. Thus, to reduce the number of iterations needed to reach the goal number of unmatched, for each iteration, we remove the edge that gives the greatest reduction in the number of unmatched, after assessing all currently remaining edges in any dimension. If several equally good options are found, the one giving the narrowest width is chosen. In Section 3.2, we describe how the widths for different covariates are scaled to be comparable.

Autostrata also provides the researcher with input parameters for more control over the resulting stratification:

- The maximum wanted numbers of unmatched treated and controls
- The maximum allowed widths between stratum edges per covariate

The researcher can specify maximum numbers of unmatched treated and controls as two separate input parameters. The stratification process will continue until reaching both numbers or until the point when there is no closer solution. For example, suppose the stratification algorithm reaches one of the requested maximum numbers of unmatched for either treated or controls. It will then continue until reaching the requested number of unmatched for the other group. It continues iterating, and the numbers can continue to improve for both treated and controls. Section 3.2 describes how Autostrata incorporates weights to account for the difference in the requested maximum numbers of unmatched treated and controls while iterating.

Further, Autostrata has a parameter for the maximum allowed stratum width per covariate, and it will not create strata wider than the given widths. If widths are not of importance, a large or infinite value can be given as input instead. The background for the maximum width parameter is that researchers may want to set a maximum difference, *caliper*, between treated and controls for the covariates—for example, max five years age difference or five points difference for a given performance score. In addition, setting a maximum width restricts the maximum imbalance. Another reason to set widths, which concerns the algorithm, is to prevent a single or a few strata from expanding too much while leaving others unchanged. Broader strata have a higher potential imbalance. It is possible to imagine that, on average, a large stratum combined with many narrow ones may somewhat cancel each other out imbalance-wise, but it is probably not what we want. A large stratum will still have a greater risk of being imbalanced. Lastly, we can use the widths produced by CEM’s histogram binning as input to Autostrata. Histogram binning only supports uniform width strata, but Autostrata can use these widths as the maximum allowed when defining strata of non-uniform widths.

3.2 Heuristics

In section 3.1, we gave an introduction to the overall approach. Autostrata is an algorithmic approach to stratifying covariates that starts with an initial state where all stratum edges are present and iteratively removes one edge at a time until the end criterion is met or no further improvements are found. Here, we describe the heuristics in more detail.

When we remove a stratum edge along the direction of one dimension (covariate), two and two strata become merged

to form new, wider strata. Removal of an edge usually results in more than two strata being merged because there are also edges along the other dimensions separating the covariate values into distinct strata. If two neighboring strata contain only treated and only controls, respectively, merging the two strata results in a stratum with a mix of both types. These units are no longer unmatched and, thus, not pruned from the sample.

Autostrata has two criteria for choosing which stratum edge to remove for each iteration. The first criterion has the highest priority, and the second criterion applies only to alternatives with equally good values for the first. The two criteria are:

1. Choose the greatest relative increase in matched treated and controls if the stratum edge is removed
2. Choose the stratum with the narrowest width

Instead of using the increase in matched units directly, Autostrata uses a weighted measure for increase. Let Δ_t and Δ_c be the increase in the number of matched treated and control units, respectively, when we remove a given stratum edge. The relative increase, Δ_{rel} , is then:

$$\Delta_{rel} = w_t \Delta_t + w_c \Delta_c$$

, where w_t and w_c are weights. The weight for the treated group, $w_t = w(t)$, and control group, $w_c = w(c)$, is found as follows:

$$w(g) = \begin{cases} \frac{m_g^{cur} - m_g^{max}}{n_g - m_g^{max}}, & m_g^{cur} - m_g^{max} \geq 0 \\ 0, & m_g^{cur} - m_g^{max} < 0 \end{cases}$$

, where g is the group, m_g^{cur} is the number of currently unmatched units for the group, m_g^{max} is the requested maximum number of unmatched for the group, and n_g is the total number of units from the group in the sample. Here, we also assume that $n_g > m_g^{max}$.

The purpose of the weights is threefold:

1. If one group is represented less than the other, each new matched unit from the group should weigh more.
2. The researcher can set parameters for how many unmatched (pruned) treated and controls are acceptable. The difference $n_g - m_g^{max}$ takes into account that the gap between available and discardable units can differ between groups.
3. If Autostrata has reached the goal for the number of unmatched units for one group, an increase in the other groups should weigh more when choosing an edge to remove. As one group comes closer to the goal, reducing the number of unmatched for the other group is prioritized higher. The difference $m_g^{cur} - m_g^{max}$ is the remaining units to match for the given group.

Width is the second selection criterion for edge removal. The widths must be scaled because Autostrata compares stratum edges from all covariates per iteration. We compute a scale factor by removing outliers and taking the min-max difference. Observations having a standard score, $|z| \geq 3$, are outliers. The data can be scaled once as an initial step. In that case, the maximum widths must be scaled as well. Also, we must restore the resulting stratum edges to the original scale. For clarity, the pseudocode in Listing 1 does not scale the data until needed.

3.3 Algorithm

Here we present the algorithm in pseudocode form. The pseudocode is at an abstraction level sufficient to implement the algorithm. However, we omit implementation details and performance enhancements that do not contribute to the understanding. Listing 1 presents the algorithm in pseudocode form, and Table 1 describes the variables used in the listing.

| Variable | Meaning |
|--|---|
| tr and ct | The covariate values for the treated and the control units |
| Δ_{best} | The best relative increase in matched units for the current iteration |
| Δ_{cur} | The relative increase in matched units for currently assessed edge |
| Δ_t and Δ_c | The increase in the number of matched treated and controls for assessed edge |
| m_t^{cur} and m_c^{cur} | The current number of unmatched treated and controls |
| m_t^{max} and m_c^{max} | The requested maximum number of unmatched treated and controls |
| covariates | The covariates (dimensions) |
| cov | The current covariate |
| edges | The current set of edges, including the outer left- and rightmost edge per covariate |
| n _{edg} | The number of edges in the current set of edges |
| edges _{cov} | The current set of edges for the current covariate, <i>excluding</i> the outer left and right edges |
| e _{cur} | The currently assessed edge |
| e _{sel} | The currently best edge for the iteration and candidate for selection |
| e _l and e _h | e _{cur} 's lower and higher adjacent edges |
| width _{cur} | The scaled widths of merged strata if we remove the currently assessed edge |
| width _{sel} | The scaled widths of strata if removing the iteration's current candidate for best edge |
| widths _{max} and width _{max} | The set of maximum allowed stratum widths, and the maximum width for the current covariate |

Table 1. The pseudocode variables and their meaning

| Autostrata Algorithm | |
|----------------------|---|
| 1 | Input: tr, ct, widths _{max} , m_t^{max} , m_c^{max} |
| 2 | Output: edges |
| 3 | Initialization of variables: |
| 4 | edges _{cov} \leftarrow one edge per distinct covariate value |
| 5 | m_t^{cur} , $m_c^{cur} \leftarrow$ calculate the initial number of unmatched treated and controls |
| 6 | Stratification: |

```

7  while ( $m_t^{cur} > m_t^{max}$  or  $m_c^{cur} > m_c^{max}$ )
   and ( $n_{edg} > 0$ ) do
8     $\Delta_{best} \leftarrow -1$ 
9     $width_{sel} \leftarrow \infty$ 
10    $e_{sel} \leftarrow \text{nil}$ 
11   for cov in covariates do
12     for  $e_{cur}$  in  $edges_{Scov}$  do
13       get  $e_l$  and  $e_h$ 
14        $width_{unscaled} \leftarrow |e_h - e_l|$ 
15       if  $width_{unscaled} > width_{max}$  then
16         continue // stratum too wide
17        $width_{cur} \leftarrow scaled\_width(e_l, e_h)$ 
18        $\Delta_t, \Delta_c \leftarrow$  the difference in numbers of
unmatched (for the multi-dimensional
strata) between  $e_l$  and  $e_h$  before and
after removing  $e_{cur}$ 
19        $\Delta_{cur} \leftarrow relative\_increase(\Delta_t, \Delta_c)$ 
20       if ( $\Delta_{cur} > \Delta_{best}$ ) or ( $\Delta_{cur} == \Delta_{best}$ 
and  $width_{cur} < width_{sel}$ ) then
21          $\Delta_{best} \leftarrow \Delta_{cur}$ 
22          $width_{sel} \leftarrow width_{cur}$ 
23          $e_{sel} \leftarrow e_{cur}$ 
24       end // if
25     end // for  $e_{cur}$ 
26   end // for cov
27   if  $e_{sel} == \text{nil}$  then
28     break // no more improvements found
29   else
30     remove  $e_{sel}$  from edges
31     update  $m_t^{cur}$  and  $m_c^{cur}$ 
32   end
33   if  $m_t^{cur} \leq m_t^{max}$  and  $m_c^{cur} \leq m_c^{max}$  then
34     break // goal reached
35   end // while
36   return edges

```

Listing 1. Pseudocode for the algorithm

3.4 Implementation

A version of the algorithm corresponding to Listing 1 was implemented in Python 3.9 (<https://python.org>), with some added performance enhancements. For example, we utilize Numba (<https://numba.pydata.org>) for counting unmatched units in strata, yielding a speedup [20, p. 125] of 2.25 for the algorithm as a whole when stratifying for Dataset 3 in Table 2 on an Intel i7-8850H CPU with 12 logical cores. A far more significant performance enhancement is achieved by caching already computed results for each stratum. The same strata are visited repeatedly during the iterations, and the algorithm finishes 17.67 times faster for Dataset 1 in Table 2 when reusing already computed results. Further, strata not affected by removing a given edge are not visited unnecessarily. Lastly, only relevant units are included in computations regarding subsets of strata.

Still, there is plenty of room to enhance performance. Many of the algorithm’s computational tasks can be performed independently, e.g., the difference in unmatched units if a given edge is removed. Such independent

computations that are well suited for parallelization are often termed embarrassingly parallel [21, p. 79-98]. A systematic approach to parallelizing algorithms is found in Foster’s methodology [22]. In addition to parallelization, we can enhance the performance by designing data structures for efficient access to frequently used data and extensively reusing previously computed results in the algorithm’s iterations. For clarity, we concentrate on the basic algorithm in this paper, leaving the suggested performance enhancements to future work.

The accompanying source code for the paper is available on GitHub (<https://github.com/jo-inge-arnes/autostrata>).

4 EXPERIMENTS AND RESULTS

4.1 Datasets

A generator for synthetic data was implemented that lets us draw random samples from a composition of distributions for treated and controls. Figure 6–Figure 8 show the datasets as violin and swarm plots, and Table 2 shows the number of units and the mixed distributions for the datasets.

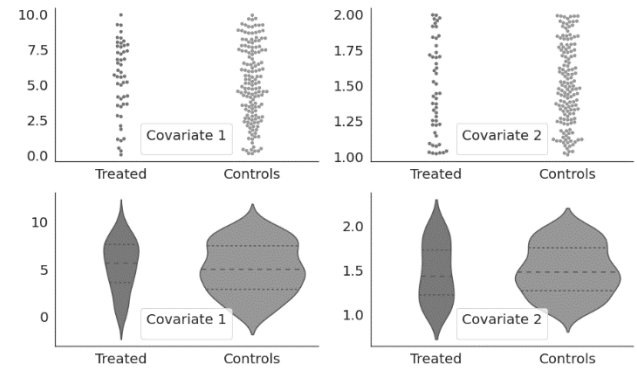


Figure 6 shows the swarm and violin plots of Dataset 1 with two uniformly distributed covariates.

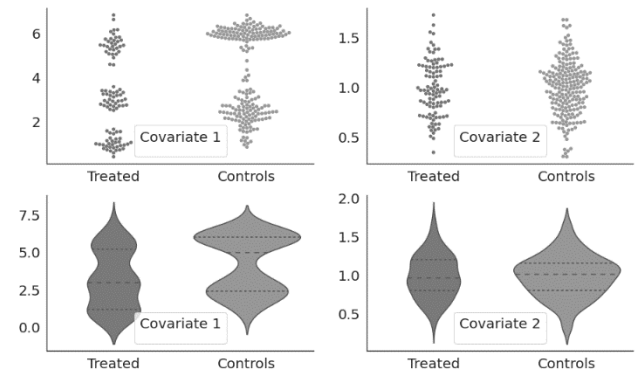


Figure 7 shows the swarm and violin plots of Dataset 2 with a mixture of Gaussians.

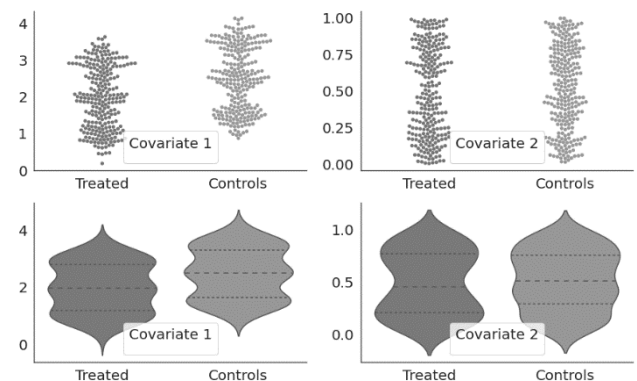


Figure 8. Dataset 3 has a mixture of Gaussians for Covariate 1 and uniform distribution for Covariate 2.

| | Treated | Controls |
|-----------|---------------------------|---------------------------|
| Dataset 1 | 50 units | 150 units |
| Cov. 1 | U(0, 10) | U(0, 10) |
| Cov. 2 | U(1, 2) | U(1, 2) |
| Dataset 2 | 100 units | 200 units |
| Cov. 1 | $N(1, \frac{1}{3.5})$ | $N(2.5, \frac{2.5}{3.5})$ |
| | $N(2, \frac{1}{3.5})$ | $N(6, \frac{1}{3.5})$ |
| | $N(5.5, \frac{1.5}{3.5})$ | – |
| Cov. 2 | $N(0.95, \frac{1}{3.5})$ | $N(1.0, \frac{1}{3.5})$ |
| Dataset 3 | 250 units | 250 units |
| Cov. 1 | $N(1, \frac{1}{3.5})$ | $N(1.5, \frac{1}{3.5})$ |
| | $N(2, \frac{1}{3.5})$ | $N(2.5, \frac{1}{3.5})$ |
| | $N(3, \frac{1}{3.5})$ | $N(3.5, \frac{1}{3.5})$ |
| Cov. 2 | U(0, 1) | U(0, 1) |

Table 2 shows the number of units and the mixed distributions for the datasets. U(min, max) stands for uniform and $N(\mu, \sigma)$ for normal distribution.

4.2 Experiments

To automate the experiments, we wrote Python and R scripts. The role of the R scripts is to call the CEM library. A reference manual for the CEM library is available online (<https://CRAN.R-project.org/package=cem>). In the code for the experiments, rpy2 (<https://rpy2.github.io>) is used to bridge between Python and R.

The experiments are as follows:

1. We call CEM to get pre-stratification scores and statistics for the given dataset.
2. Next, CEM is used to stratify the covariates by applying Scott’s rule for histogram binning. It also computes the number of unmatched units, imbalance scores, and other statistics.
3. We then pass CEM’s outputted number of unmatched units and stratum widths to Autostrata.
4. Autostrata stratifies the covariates.
5. Autostrata’s outputted stratum edges are given as input to CEM, which uses them to stratify and compute statistics equivalent to step 2.

Two experiments are conducted per dataset. They differ only in how the results are passed to Autostrata in Step 3:

| Input type | Input parameters |
|------------|--|
| P1 | The numbers of unmatched treated and controls from histogram binning are passed as m_t^{max} and m_c^{max} and the bin widths are passed as $widths_{max}$. |
| P2 | The m_t^{max} and m_c^{max} values are as in P1, but $widths_{max}$ values are set to infinity. |

Table 3. Input parameters. See Table 1 for variables

4.3 Results

Table 4 shows the experiment results.

DS1, DS2, and DS3 are headers for the results of the three datasets. The top column headers stand for ‘results before stratification’ (Before), ‘stratification with histogram binning’ (Hist.), and the input types P1 and P2 from Table 3. ‘Res.’ is an abbreviation for results, and ‘Imp.’ is the percent improvement compared to histogram binning.

The row labels denote multivariate imbalance measure (MIM), total unmatched (UM_{TOT}), unmatched treated (UM_{TR}), and unmatched controls (UM_{CT}).

TOI is the percent improvement in the trade-off, which is the sum of the improvements for UM_{TOT} and MIM.

| | Before | Hist. | P1 | | P2 | |
|------------|--------|-------|-------|-----------|-------|------------|
| | Res. | Res. | Res. | Imp. | Res. | Imp. |
| DS1 | | | | | | |
| MIM | 0.240 | 0.199 | 0.167 | 16% | 0.187 | 6% |
| UM_{TOT} | 0 | 65 | 80 | -23% | 54 | 17% |
| UM_{TR} | 0 | 4 | 5 | -25% | 3 | 25% |
| UM_{CT} | 0 | 61 | 75 | -23% | 51 | 16% |
| TOI | – | – | – | -7% | – | 23% |
| DS2 | | | | | | |
| MIM | 0.465 | 0.273 | 0.266 | 3% | 0.335 | -23% |
| UM_{TOT} | 0 | 88 | 84 | 5% | 72 | 18% |
| UM_{TR} | 0 | 16 | 24 | -50% | 16 | 0% |
| UM_{CT} | 0 | 72 | 60 | 17% | 56 | 22% |
| TOI | – | – | – | 7% | – | -5% |
| DS3 | | | | | | |
| MIM | 0.348 | 0.290 | 0.174 | 40% | 0.280 | 3% |
| UM_{TOT} | 0 | 53 | 92 | -74% | 39 | 26% |
| UM_{TR} | 0 | 21 | 53 | - | 21 | 0% |
| | | | | 152% | | |
| UM_{CT} | 0 | 32 | 39 | -22% | 18 | 44% |
| TOI | – | – | – | -34% | – | 30% |

Table 4. Results from experiments. Best TOI results per dataset are in bold and thicker cell borders.

5 DISCUSSION

Table 4 shows that both imbalance and the total number of unmatched units are lower for Autostrata for all three datasets. The input parameter type P2 gave the best results for DS1 and DS3, while P1 gave the best for DS2. The difference is that P2 sets the maximum allowed stratum widths to infinity, which effectively disables the parameter. By visually comparing the swarm plots in Figure 6–Figure 8, we see the difference between DS2 and the other two: DS2 has several regions with minimal overlap between treated and controls. As Section 3.1 explains, finding narrow strata with mixed types of units is easier in regions with high overlap. Therefore, restricting the widths is usually not necessary in such regions. Autostrata also works well for sparser overlap, but as illustrated by the experiment for DS2, setting maximum widths is more important.

Autostrata competed with CEM’s best effort in the experiments, and we passed parameters not necessarily ideal for non-uniform widths. It is possible to adjust these parameters manually or programmatically, but for objectivity, we use the unchanged output from CEM as input to Autostrata.

Lastly, Autostrata can be used stand-alone. A researcher can decide the acceptable differences between treated and controls based on domain knowledge. The researcher can also request a maximum number of unmatched units. Autostrata thus provides researchers with more up-front control. After stratification, the researcher can input the stratum edges to the CEM software as manual cutpoints. A

combination is even possible, where Autostrata stratifies a subset of the covariates given to CEM.

6 RELATED WORK

Aikens, R.C. et al. [23] present *Stratamatch*, a method for stratification of covariates for CEM. Only datasets from a minimum of 5 000 up to millions of observations are recommended. The method divides the dataset into training (pilot) and analysis sets, and the resulting strata are close to equal-sized. The size must be manually decided.

Jackson, B. et al. [24] present an algorithm for optimal data partitioning on an interval that Scargle, J.D. et al. [25] apply for astronomical time series. The algorithm supports custom fitness functions, and we tried defining a function. However, a common issue is the unwanted case of one subinterval per value; thus, the researcher must choose an expected number of subintervals. Also, while theoretically possible to extend for multivariate data, the algorithm is primarily univariate.

7 CONCLUSION

We have presented Autostrata, an algorithmic approach to stratifying confounder covariates. Autostrata shows improved results compared to the standard CEM stratification. In addition, it provides the researcher with parameters for controlling the stratification. Autostrata can be used stand-alone.

8 REFERENCE

- [1] Ranganathan, P., Aggarwal, R. “Study Designs: Part 3 – Analytical Observational Studies” in *Perspectives in Clinical Research*, Vol. 10, Issue 2, pp. 91-94. 2019.
- [2] Hariton, E., Locascio, J.J. “Randomised Controlled Trials – The Gold Standard for Effectiveness Research” in *BJOG: An International Journal of Obstetrics & Gynaecology*, Vol. 125, Issue 13, pp. 1716-1716. 2018.
- [3] Arnes, J.I., Bongo, L.A. “The Beauty of Complex Designs” in *Advancing Systems Epidemiology in Cancer: Exploring Trajectories of Gene Expression*, pp. 23-47. Scandinavian University Press, 2020.
- [4] Kim, S. “Case-Cohort Studies vs Nested Case-Control Studies” in *Datum Newsletter Division of Biostatistics*, Vol. 22, Issue 1, pp. 1-2. 2016.
- [5] Ngo, L.H., et al. “Methodologic Considerations in the Design and Analysis of Nested Case-Control Studies: Association Between Cytokines and Postoperative Delirium” in *BMC Medical Research Methodology*, Vol. 17, Issue 1, pp. 88. 2017.
- [6] Alexander, L.K., et al. “Confounding Bias, Part I” in *ERIC Notebook*, Vol. 11. 2015.
- [7] Alexander, L.K., et al. “Confounding Bias, Part II and Effect Measure Modification” in *ERIC Notebook*, Vol. 12. 2015.
- [8] Howards, P.P. “An Overview of Confounding. Part 1: The Concept and How to Address It” in *Acta Obstetrica et Gynecologica Scandinavica*, Vol. 97, Issue 4, pp. 394-399. 2018.
- [9] Wacholder, S., et al. “Selection of Controls in Case-Control Studies: I. Principles” in *American Journal of Epidemiology*, Vol. 135, Issue 9, pp. 1019-1028. 1992.
- [10] Tripepi, G., et al. “Stratification for Confounding – Part 1: The Mantel-Haenszel Formula” in *Nephron*

Clinical Practice, Vol. 116, Issue 4, pp. 317-321. 2010.

- [11] Iacus, S.M., King, G., Porro, G. “Matching for Causal Inference Without Balance Checking” in *SSRN Electronic Journal*. 2008.
- [12] Blackwell, M., et al. “cem: Coarsened Exact Matching in Stata” in *The Stata Journal*, Vol. 9, Issue 4, pp. 524-546. 2009.
- [13] Iacus, S.M., King, G., Porro, G. “cem: Software for Coarsened Exact Matching” in *Journal of Statistical Software*, Vol. 30, Issue 9, pp. 1-27. 2009.
- [14] Sturges, H.A. “The Choice of a Class Interval” in *Journal of the American Statistical Association*, Vol. 21, Issue 153, pp. 65-66. 1926.
- [15] Scott, D.W. “On Optimal and Data-Based Histograms” in *Biometrika*, Vol. 66, Issue 3, pp. 605-610. 1979.
- [16] Freedman, D., Diaconis, P. “On the Histogram as a Density Estimator: L2 theory” in *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, Vol. 57, Issue 4, pp. 453-476. 1981.
- [17] Shimazaki, H., Shinomoto, S. “A Method for Selecting the Bin Size of a Time Histogram” in *Neural Computation*, Vol. 19, Issue 6, pp. 1503-1527. 2007.
- [18] King, G., Nielsen, R. “Why Propensity Scores Should Not Be Used for Matching” in *Political Analysis*, Vol. 27, Issue 4, pp. 435-454. 2019.
- [19] Sekhon, J.S. “The Neyman-Rubin Model of Causal Inference and Estimation via Matching Methods” in *The Oxford Handbook of Political Methodology*, Vol. 2, pp. 1-32. 2008.
- [20] Pacheco, P.S. *An Introduction to Parallel Programming*. Morgan Kaufmann, Burlington, MA, 2011.
- [21] Wilkinson, B., Allen, M. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. Pearson/Prentice Hall, Upper Saddle River, N.J, 2005.
- [22] Foster, I. *Designing and Building Parallel Programs*. Addison-Wesley, Reading, MA, 1995. Also available from: <https://www.mcs.anl.gov/~itf/dbpp/> Accessed 2022-06-26.
- [23] Aikens, R.C., et al. “stratamatch: Prognostic Score Stratification Using a Pilot Design” in *arXiv preprint arXiv:2001.02775*. 2020.
- [24] Jackson, B., et al. “An Algorithm for Optimal Partitioning of Data on an Interval” in *IEEE Signal Processing Letters*, Vol. 12, Issue 2, pp. 105-108. 2005.
- [25] Scargle, J.D., et al. “Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations” in *Astrophysical Journal*, Vol. 764, Issue 2, pp. 167. 2013.

9 ACKNOWLEDGEMENT

We want to thank Marc Weitz for inspiring the use of swarm and violin plots.

Paper III

Greedy Knot Selection Algorithm for Restricted Cubic Spline Regression

Jo Inge Arnes, Alexander Hapfelmeier, Alexander Horsch, and Tonje Braaten

In submission. Preprint DOI: [10.21203/rs.3.rs-2708178/v1](https://doi.org/10.21203/rs.3.rs-2708178/v1)

Greedy Knot Selection Algorithm for Restricted Cubic Spline Regression

Jo Inge Arnes^{1*}, Alexander Hapfelmeier², Alexander Horsch¹ and Tonje Braaten³

¹Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway.

²Institute of AI and Informatics in Medicine, Technical University of Munich, München, Germany.

³Department of Community Medicine, UiT The Arctic University of Norway, Tromsø, Norway.

*Corresponding author(s). E-mail(s): jo.i.arnes@uit.no;

Contributing authors: alexander.hapfelmeier@mri.tum.de; alexander.horsch@uit.no;
tonje.braaten@uit.no;

Abstract

Non-linear regression modeling is common in many fields for prediction purposes or estimating relationships between predictor and response variables. For example, restricted cubic spline regression can model non-linear relationships as third-order polynomials joined at knot points. The standard approach is to place knots by a regular sequence of quantiles between the outer boundaries. A regression curve can easily be fitted to the sample using a relatively high number of knots. The problem is then overfitting, where a regression model has a good fit to the given sample but does not generalize well to other samples. A low knot count is therefore preferred. However, the standard knot selection process can lead to underperformance in the sparser regions of the predictor variable, especially when using a low number of knots. It can also lead to overfitting in the denser regions. We present a simple greedy search algorithm using a backward method for knot selection that shows reduced prediction error and Bayesian information criterion (BIC) scores compared to the standard knot selection process in simulation experiments. We have implemented the algorithm as part of an open-source R-package, **knutar**.

Keywords: model selection, non-linear regression, prediction, restricted cubic splines

1 Introduction

Regression modeling is used in many fields for prediction purposes or for estimating relationships between predictor and response variables. For example, we may be interested in studying the relationship between explanatory variables and outcomes in fields such as epidemiology, biostatistics, clinical research, economics, and psychology.

As a starting point, such relationships can be assumed to be linear, but when the assumption does not hold, non-linear methods can be employed. It has often been questioned if a single correct model even exists for a non-linear prediction problem (Chatfield, 1995; Steyerberg, 2019). Instead, multiple alternatives may be useful (Gauthier et al, 2020).

One alternative is to use *restricted cubic spline* (RCS) regression (Harrell, 2015, pp. 23-26), which involves partitioning the observations of a predictor variable into subintervals and piecewise fitting a third-order polynomial to each subinterval. The splines connect at join points called knots, and the RCS regression method ensures the overall function's smoothness by forcing the first and second derivatives of the connected polynomials to agree at the knots. It additionally restricts splines to be linear in the tails of the boundary knots because unrestricted splines tend to behave poorly at the boundaries of the data (Perperoglou et al, 2019, p. 6). RCS regression models comprise simple polynomial functions that are well-suited for interpretation by the researcher and can be combined with widely used analysis models. As Buis (2009) states, "restricted cubic splines are an easy way of including an explanatory variable in a smooth non-linear way in a wide variety of models."

The standard process for placing knots for RCS regression is by a regular sequence of quantiles for the observed values of the predictor variable between two boundary knots. These boundary knots are often placed at the 0.05 and 0.95 percentiles. Recommendations for knot counts, quantiles, and boundaries are found in (Harrell, 2015, pp. 27-28). In this paper, we use 0.05 and 0.95 percentiles as the outer boundaries for the predictor variable observations. Fig. 1 shows the curve approximated by an RCS regression model fitted to a sample where the relationship between the predictor and response variable is non-linear.

Placing knots based on quantiles is an accepted convention, but there are, for example, rarely any biological or other reasons dictating that the relationship between predictor and response variables must align with equal-sized quantiles between the boundary knots. Nevertheless, there are several reasons for using a regular sequence of quantiles. For example, such quantiles are separated by the same number of observations, ensuring that observations exist between each pair of knots. In contrast, the same is not true for equidistant knots, which can lead to empty subintervals or non-convergence of the model's fitting procedure. Further, if certain subintervals have too few observations, it can result in instability of estimates where the estimates become very sensitive to the specific values in these regions.

Using many quantiles results in knots being close to each other, especially in the denser regions of the predictor variable's distribution. If the knots are sufficiently close, spline regressions can readily fit a model to the sample data. As the number of knots increases, the degrees of freedom and the complexity of the model increase. The problem is then the risk of overfitting, which means that with increasing model complexity, the models will often match the given sample better but not the other samples from the data-generating process or population in general. Therefore, keeping the number of knots and the respective model complexity low is desirable, yielding models that fit a given sample less exactly but generalize better. A knot count of five or less is usually considered sufficient in practice (Harrell, 2015; Gauthier et al, 2020; Stone, 1986). Using five knots is a good choice when the sample size is large, $n \geq 100$, for continuous uncensored response variables, according to (Harrell, 2015, p. 28).

Several measures for estimating the goodness of fit of a model exist that penalize higher knot counts, such as Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) (Schwarz, 1978; Zucchini, 2000). Unfortunately, when we are limited to placing knots by a regular sequence of quantiles, having a low number of knots may miss locations essential for a good model fit. Furthermore, we risk placing knots in locations that do not substantially improve the fit or can contribute to overfitting in denser subintervals.

For spline regression models, in general, the number of knots and locations are hyperparameters that must be chosen. Perperoglou et al (2019) describe the role of spline regression models in modern biostatistics and review software packages for spline functions in R (R Core Team, 2022). The paper was written on behalf of the STRengthening Analytical Thinking for Observational Studies (STRATOS) initiative (Sauerbrei et al, 2014). The authors conclude that an experienced analyst can achieve reasonable outcomes, regardless of the spline type or tool. Most differences can be attributed to the choice of hyperparameters. However, analysts may not possess sufficient knowledge, and the availability of user-friendly, well-documented software packages for spline modeling is identified as important.

Against this background, we present a knot selection process for RCS regression models of low complexity. The process empirically shows improved results compared to placing knots separated by equal-sized quantiles for comparable knot counts. The algorithm is implemented as part of a software package for R, `knutar`. In addition to RCS, the function `choose_model` uses fractional polynomial (FP) regression (Royston and Altman, 1994). It selects a model based on the best goodness of fit from either FP regression, RCS with equal-sized quantiles, or RCS using the knot selection process presented in this paper.

The rest of the paper is organized as follows: Section 2 presents the novel knot selection process and algorithm. Section 3 describes the data generator designed for generating artificial datasets for simulation experiments. The software package with the implementation of the knot selection process and the source code repository for experiments are covered in Section 4. Section 5 describes the method for the experiments, followed by experiments and results in Section 6. The discussion is found in Section 7. Notable related work is briefly described in Section 8 before concluding in Section 9. Lastly, Section 11 declares that the authors have no competing interests related to the work.

2 Knot selection process

This section presents a process for placing knots for RCS regression. The process is an algorithm that finds a model with a good fit but an inflated number of knots and uses its knot locations to search for a less complex model. We first provide a justification of the process before describing the algorithm in more detail in sections 2.1 and 2.2.

For the rest of the paper, we distinguish between inner and boundary knots. The *inner* knots are all knots except the two boundary knots.

The standard process for placing inner knots is by a regular sequence of quantiles between the boundaries, which leaves only a single way of placing k inner knots for a given sample. Alternatively, we could allow the inner knots to be placed freely. For freely placed knots, the number of possible ways to place k inner knots for a sample theoretically becomes infinite for a real predictor variable, $x \in \mathbb{R}$. In practice, it is unnecessary to place knots

indefinitely close. Beyond a certain level of precision in identifying knot locations, the differences in model fit become so small that they are no longer practically meaningful or valuable, i.e., a limited accuracy in closeness is sufficient. Consequently, there is a finite set of q locations where inner knots can be placed in the interval for the predictor variable observations. The total number of combinations that $r = k$ inner knots can be placed for q locations is equal to the binomial coefficient:

$$C(q, r) = \binom{q}{r} = \frac{q!}{r!(q-r)!}. \quad (1)$$

The model found by the standard knot selection process is only one in a more extensive set of possible models in the same model family. Therefore, it seems likely that other models in the same family having a better fit exist.

By reducing the accuracy for the closeness of the candidate knot locations, the q number of locations available for knot placement becomes lower. Simultaneously, the number of possible ways to arrange the r inner knots drops substantially, as understood from the binomial coefficient. The set of locations available for knot placement can also be interpreted as a partitioning of the predictor variable interval. The knot selection process in this paper finds a reasonable partitioning of the interval where r inner knots are to be placed. It defines a manageable number of locations, $q \geq r$, where inner knots may freely be placed and places knots so that the resulting model yields a low BIC score. In this paper, the partitioning is taken from the knot locations of a model with $q \geq r$ inner knots, often having tens of knots, fitted to the sample by applying the standard knot selection process.

The following subsections present the process as an algorithm with two main steps. The first step of the algorithm, described in Section 2.1, is to find a suitable start model. The start model's inner knots define all the q locations where the r inner knots of a final model may possibly be placed. The second step of the algorithm starts from the full set of knot locations in the starting model and removes knot locations one by one iteratively. The aim is to find a model with r inner knots that is better than the model obtained from the standard process directly.

The strategy behind the algorithm is comparable to *backward methods* that start with

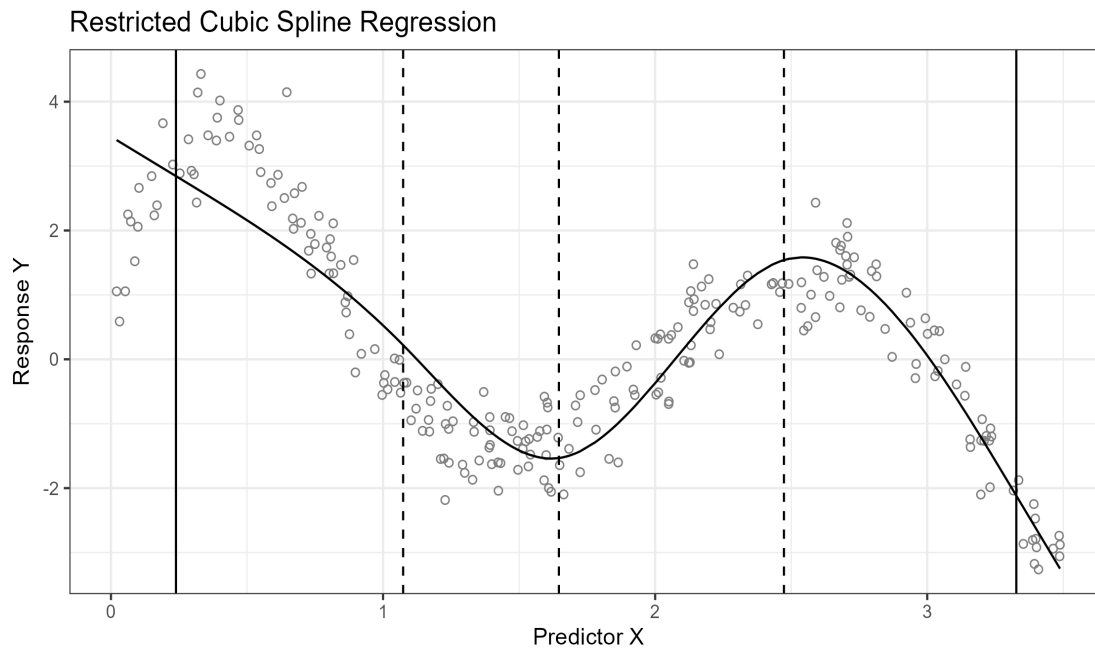


Fig. 1 The figure shows the curve approximated by a five-knot RCS regression model fitted to a sample of 250 observations. The predictor, X , and response, Y , variables have a non-linear relationship. The inner knot locations are marked with dashed vertical lines along the horizontal axis, and the 0.05 and 0.95 percentile boundary knots are marked with solid vertical lines. The knots have been placed using the standard knot selection process, where the inner knots are placed by a regular sequence of quantiles between the boundary knots. The observations are shown as gray circles. The knots partition the observations into subintervals having 56 observations each, and there are 13 observations outside each boundary knot. The predictor variable is uniformly distributed, $X \sim U(0, 3.5)$, and the response variable values are the sum of a fifth-degree polynomial function, $X(X - 1)(X - 2)(X - 3)(X - 4)$, representing the true curve, and a stochastic error term, $E \sim \mathcal{N}(0, 0.5)$, representing the (homoscedastic) variance.

complicated models, such as a high-degree polynomial, and successively simplifies them (Kleinbaum, 2014, p. 48). It is distinct from *backward elimination* strategies that remove variables from a set of study variables (Kleinbaum, 2014, p. 172).

2.1 Finding a start model

The first step of the algorithm searches for a suitable start model by comparing the fits of a series of models up to a relatively high knot count, for example, $k = 0, \dots, 50$, where k is the number of inner knots. The knot count does not include the outer boundaries. A quantitative criterion for estimating the model's goodness of fit, which additionally considers the knot count, i.e., penalizes for model complexity, is used to compare the models. AIC is one such criterion that can be used. BIC is another (Schwarz, 1978; Zucchini, 2000). We have chosen to use BIC because it more strongly penalizes higher knot counts than AIC. Furthermore, the BIC is an asymptotically consistent model selection criterion, meaning that it almost surely, with a probability approaching one with $n \rightarrow \infty$,

selects the correct model from a family containing this model (Hastie et al, 2009, p. 235), where n is the sample size. For both AIC and BIC, lower scores indicate better goodness of fit, adjusted for model complexity. We select the model yielding the lowest BIC score of the assessed models as our start model.

Figure 2 shows the BIC scores for a set of models with different numbers of inner knots, $k = 0, \dots, 50$. Each model is fitted to the same sample by RCS regression and the standard procedure for knot selection. In the figure, the inner knot counts are shown along the horizontal axis, and the BIC scores along the vertical axis. The figure illustrates that increasing the number of knots typically leads to progressively lower BICs, possibly having local minima, before reaching a global minimum. Beyond this number of knots, the BIC scores increase. The curve shows how BIC reflects that overly simple or complex models have the propensity to underperform. Well-known reasons are that they may not be able to cover relations well enough or may fit to the noise, respectively. Zucchini (2000) describes the former

as *discrepancy due to approximation*, the latter as *discrepancy due to estimation*, and the combination as the *expected (overall) discrepancy*. The discrepancy due to estimation increases as the number of knots increases. This may be less severe for larger samples, e.g., having thousands of observations. In that case, the increase in overall discrepancy becomes less steep. In this paper, the focus is on hundreds of observations per sample rather than thousands.

Given the low BIC, the start model has a relatively low expected (overall) discrepancy. However, it has far more knots than the maximum we want for the final model. It seems probable that the start model is overfitted but that a subset of the knot locations can be a good choice for building a less complex model.

2.2 Lowering the model complexity

The start model has the lowest BIC score of all assessed models found by applying the standard knot selection process for a range of knot counts and, presumably, a low expected (overall) discrepancy. Next, the second step of the algorithm uses this model as a starting point for iteratively searching for a less complex model while keeping the BIC low.

Let q be the number of inner knots in the start model, and r be the target knot count of the less complex model. The value of q depends on the first step of the algorithm, and r is decided up front by the analyst and fixed. Let \mathfrak{S} be the set of locations of the inner knots in the start model.

A possible approach to finding a less complex model is to compare all possible models having inner knots at r locations selected from \mathfrak{S} . We can then perform an exhaustive, brute-force search and be guaranteed to find the best model in this candidate set, \mathfrak{B} . Unfortunately, the size of \mathfrak{B} rapidly grows as q increases. We observe that the number of models in the set equals the binomial coefficient, $|\mathfrak{B}| = C(q, r)$ (Equation 1), meaning that an exhaustive search does not scale well from a computational standpoint. In Θ -notation (Cormen et al, 2022, pp. 48-49), it has a factorial time complexity, $\Theta(q!)$, holding r fixed.

Here, we present a *greedy algorithm* (Cormen et al, 2022) that starts with the complete set of knot locations from the start model, \mathfrak{S} , selected. It

then removes knot locations one by one. The algorithm is a state space search (Russell and Norvig, 2010, p. 67) using a simple heuristic: Identify and remove the knot location with the least undesirable impact on the BIC score when removed. This knot location is deemed the most redundant in the current set, which implies an assumption that the individual knot locations in a model can be ordered by their relative contribution to a good fit, from being crucial to redundant. Note that it only matters which knot location is the *most* redundant for each iteration step. Only the most redundant knot location is removed. As long as none of the r knot locations for the globally best model in \mathfrak{B} have been ranked as the *most* redundant in an iteration step, the found model will be identical to the globally best model in \mathfrak{B} . When $r = q - 1$, the state space search algorithm is equivalent to assessing all models in \mathfrak{B} . Otherwise, the resulting final model may differ from the best model in \mathfrak{B} . Thus, the algorithm does not guarantee finding the best model in \mathfrak{B} . However, it is relatively common for state space search, or machine learning algorithms in general, not to guarantee a globally optimal solution.

In the first iteration step, the algorithm assesses which one of the start model's q inner knot locations to remove first. For this, it assesses q candidate models. In the next step, the algorithm assesses which one of the $q - 1$ inner knot locations of the model found in the previous step to remove next, which requires the assessment of $q - 1$ candidates. The iteration continues until r inner knot locations are left. For the last step, $r + 1$ models are assessed. Thus, the total number of models assessed by the algorithm is the sum of natural numbers from $r + 1$ to q inclusive. Here, we assume that $r < q$. By applying Equation 2 for $a = r + 1$ and $b = q$ and expanding, we get Equation 3.

$$\sum_{j=a}^b j \equiv \frac{(a+b)(b-a+1)}{2} \quad (2)$$

$$\sum_{j=r+1}^q j \equiv \frac{q+(r+1)}{2} (q-r) \equiv \frac{1}{2}(q^2 + q - r^2 - r) \quad (3)$$

From the right-hand side of Equation 3, we see that the time complexity is quadratic, $\Theta(q^2)$,

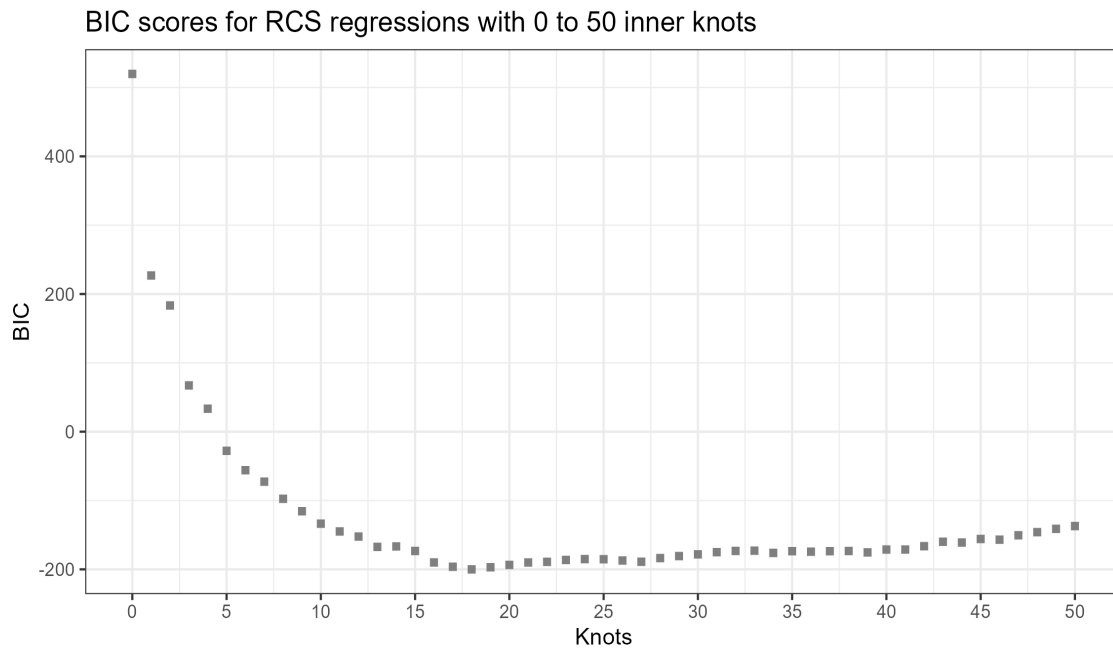


Fig. 2 BIC scores for increasing number of knots. The lowest BIC score is at 18 inner knots in this example. The sample consists of $n = 250$ observations with $\text{Lognormal}(0.5, 0.35)$ distribution and the true function is $\cos(\pi X)$. The models have 0.05 and 0.95 percentile boundary knots for the predictor variable.

holding r fixed. From equations 1 and 3, we see that for $r \leq 2$, the exhaustive search requires fewer models to be assessed than the state space search algorithm. The two approaches are equivalent when $r = q - 1$. Otherwise, the number of models assessed by the exhaustive search grows far more rapidly with increasing q . For example, if the start model has $q = 30$ inner knots and the final model has $r = 4$ inner knots, the exhaustive search assesses 27,405 models. The version that removes one knot at a time only assesses 455 models. This difference is monotonically and steeply increasing as q increases. Further, instead of allowing only a specific number of knots for the final model, we can accept a final model having a knot count in a given range. When assessing a sequence of allowed number of inner knots, $k = 0, \dots, k_{max}$, removing one knot at a time is computationally cheaper because the exhaustive search requires the complete set of possible models per target knot count, $\mathfrak{B}_k, k = 0, \dots, k_{max}$, for each step. In conclusion, the state space search scales better computationally. A combination of the exhaustive and the state space search can be used. For example, the exhaustive search can find the best model with regard to BIC in \mathfrak{B} when $r \leq 2$. However, the experiments described in sections 5–6 only use the state space search because it is the paper’s primary focus.

We may be tempted to remove all but the r most crucial inner knots in one step, but this approach can be suboptimal. When a single knot is removed, the order of the remaining inner knots by relative importance can change. A hypothetical example is when two or more inner knots are clustered around a location crucial to a good fit, e.g., an essential critical point. Individually removing any of these inner knots may have a low negative impact because the other inner knots still support the shape of the curve. On the other hand, if we remove all these inner knots, no knots would contribute to the shape of the curve around the crucial location. The consequence would be a significant negative impact on the goodness of fit. When we remove knots one by one, we allow the order of the knots by relative importance to change per iteration step. For example, if only one of the knots around a crucial location is still present in the set, its importance will be ranked as high, preventing it from being removed.

Finally, we could also imagine a different greedy algorithm that starts with no inner knots and iteratively adds knots at locations selected from \mathfrak{S} . Each iteration step adds a new knot by selecting from \mathfrak{S} the knot location that yields the best model of the alternatives. When the algorithm selects the first knot location and fits a model with a single inner knot, the discrepancy

due to approximation can be expected to be high in many cases. Thus, the selected knot location may not be essential to recreate the start model's basic regression curve shape. Unfortunately, the algorithm will not replace knot locations in subsequent iterations steps, meaning new knots can be placed based on misselected locations from the early steps.

2.3 Accepting a range of knot counts

In the previous section (Section 2.2), we described the algorithm as targeting one specific number of knots for the final model. However, we do not target only one specific number of knots in the software package (Section 4), experiment methods (Section 5), and experiments (Section 6). Instead, the final knot count is allowed to be within a range, and the model yielding the best BIC score in that range is selected as the final model. We also do the same for the standard knot selection process and select the model having the best BIC score within the given range of inner knot counts, $k = 0, \dots, k_{max}$.

To find a final model having an inner knot count within a target range $k = 0, \dots, k_{max}$ using the algorithm presented in this paper, the iteration described in Section 2.2 continues until all inner knots have been removed. The two boundary knots are never removed. The algorithm selects as the final model the best of the models found during the last iteration steps, where $k \leq k_{max}$.

2.4 Knot removal example

Figure 3 shows the effect of the algorithm iteratively removing knot locations from the start model in the search for a final model of lower complexity. The resulting regression curves at four different steps are plotted. The predictor variable is lognormally distributed, $X \sim \text{Lognormal}(0.5, 0.35)$, and the true curve is a cosine function, $\cos(\pi X)$. The algorithm finds a start model by applying the standard knot selection process for a range of inner knot counts, $k = 0, \dots, 50$. The model at 18 inner knots yields the best BIC for this sample. The algorithm then systematically removes knot locations from this start model. At each iteration step, the algorithm assesses all models with j of $j+1$ inner knot locations from the previous step, i.e., $j+1$ models, and selects the model yielding the lowest BIC score.

The figure shows the best models for 18, 13, 8, and 3 inner knots, having BIC scores of -200, -227, -247, and -186, respectively. Each step does not necessarily have a lower BIC score than the previous step because the BIC score can increase as the number of knots decreases because of discrepancy due to approximation (underfitting).

Notice that the start model with 18 inner knots in Fig. 3 has many redundant knots. As the algorithm removes such knots, the basic shape of the regression curve stays relatively stable for this example. Also, the start model's curve has some wiggleness around the top turning point. It is an example of overfitting in a region where the distribution of the observations for X is denser and illustrates one problem of placing knots by a regular sequence of quantiles.

Fig. 4 shows the predicted curve for the final model in Fig. 3 and the corresponding model obtained by applying the standard knot selection process for three inner knots directly, yielding a BIC of 67. Notice that the observations are more sparsely distributed for greater values of X , making the upper quantile before the boundary in the right-side plot of Fig. 4 wider and the model underfitted. This underfitting illustrates another problem concerning the standard knot selection process.

3 Data generator

A data generator was designed and implemented for generating the artificial datasets used in the experiments. It produces pseudo-random samples by applying three user-defined functions representing:

1. The distribution \mathcal{X} of the predictor variable $X = \{x_1, \dots, x_n\} \sim \mathcal{X}_n$, where n is the sample size. For example, the predictor variable distribution can be $\mathcal{X} = \text{Lognormal}(\mu, \sigma)$.
2. The ground truth function, ϕ , for generating the population means, F , for the response variable Y given X . The term *true curve* is used in the paper for these population means:

$$F = \mu_{Y|X} = \phi(X). \quad (4)$$

3. The distribution for the error component, \mathcal{E} , around $\mu_{Y|X}$. For example, it can be a normal distribution:

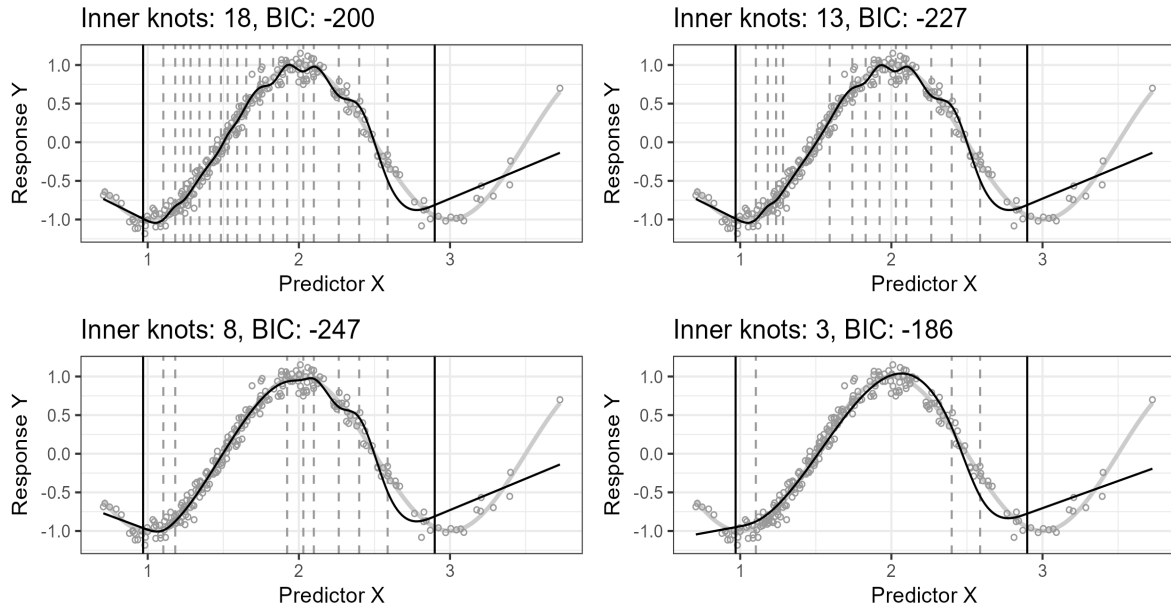


Fig. 3 The figure shows the effect of iteratively removing knot locations one by one from the start model. In each of the four plots, the thin, black line is the model's fitted curve, whereas the thicker, light gray line is the true curve. The smaller circles are the observations. The inner knot locations are shown as dashed vertical lines, whereas the 0.05 and 0.95 percentile boundary knot locations are shown as solid, black vertical lines.

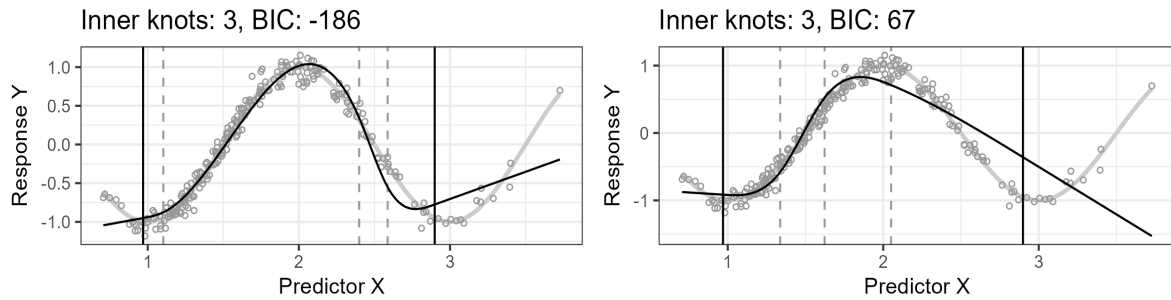


Fig. 4 The plot to the left shows the curve for the final model in Fig. 3. The plot to the right shows the curve for the model obtained by using the standard knot selection process for three inner knots directly.

$$\mathcal{E} \sim \mathcal{N}(\mu_{\mathcal{E}}, \sigma_{\mathcal{E}}). \quad (5)$$

First, the data generator draws a sample of n predictor variable observations, $X \sim \mathcal{X}_n$. For repeatability, the user can optionally set the seed used internally by the pseudo-random number generator. Next, the generator computes the true curve values, $F = \phi(X)$. The generator then computes the response variable values, $Y = F + (E \sim \mathcal{E}_n)$. The error component distribution \mathcal{E} is user-defined and can alternatively be heteroscedastic. For example, we can scale the error distribution's variance by a factor of the given value of $x \in X$, which can be relevant for ratio-valued variables. Finally, the sample values can be rounded to a chosen accuracy, simulating the limits of

the measurement method. The resulting dataset includes both the rounded and unrounded values. The simulation experiments in this paper only use homoscedasticity and unrounded values to avoid unnecessary complexity.

X and Y are the input predictor and response variable values used for fitting the model, whereas F are the ground truth values used to assess the performance of the models' predictions in the experiments. Together, they form the variable Z having the distribution \mathcal{Z}_n :

$$Z = (F, X, Y) \sim \mathcal{Z}_n \quad (6)$$

4 Software package implementation

The implementation of the knot selection algorithm presented in this paper is included as part of our package, `knutar`, for use with R (R Core Team, 2022). The package is publicly available at <https://github.com/jo-inge-arnes/knutar> and contains functions for suggesting models and utility and plotting functions. It also includes the function `generate_data` for generating artificial datasets.

The function `choose_model` in the package assesses different regression models from a set of regression methods, returning the one yielding the best results according to an information criterion, where the default information criterion is BIC.

The function's strategy for choosing the appropriate regression model follows a forward method going from simple to more complicated methods. The `knutar` package uses the standard generalized linear regression models (GLM) function in R, `stats::glm`, for building models. It first applies regressions with multivariable fractional polynomials, `mfp` (Ambler and Benner, 2022), which internally uses a forward selection process for fractional polynomials (Royston and Altman, 1994) that includes simple linear regression. Next, the function uses RCS regression, `splines::ns`, and finds the number of knots, $k \leq k_{max}$, yielding the best score for equal-sized quantiles. Lastly, the function applies the knot selection process presented in the paper. The model with the best information criterion score is returned along with extra information. If models from different methods give the same best score, the function chooses the model stemming from the earliest of the applied methods. The function additionally returns a list with the best candidate model from each of the three regression methods and information about the chosen hyperparameters.

The main parameters of `choose_model` are the dataset for the sample, the response variable, and the predictor variable(s). The function uses the response and predictor variables as the left- and right-hand sides of a formula so that the predictor variable can be a formula composed of one or more of the variables available in the dataset. Additionally, `choose_model` provides optional input parameters to replace the default

information criterion, maximum number of knots, and more.

The R-scripts for running the paper's accompanying experiments are in a separate repository at <https://github.com/jo-inge-arnes/knutar-experiments>.

5 Methods

We conduct four simulation experiments as application examples of different functions, ϕ , for the true curve. The design of the simulation experiments follows the theoretical framework for inference problems in benchmark experiments presented in (Hothorn et al, 2005).

5.1 Simulation experiment

Using the data generator described in Section 3, artificial observations are generated by drawing from known distributions. The ground truth is known. Each $z \in Z \sim \mathcal{Z}_n$ consists of the value for the predictor variable, x , the response variable, y , and the ground truth that is to be predicted, $f = \mu_{y|x}$:

$$z = (f, x, y) \quad (7)$$

For each simulation experiment, we generate artificial data with a defined distribution, \mathcal{Z} , and draw a set of $M = 1,000$ learning samples consisting of $n = 250$ observations:

$$\mathcal{L}^1, \dots, \mathcal{L}^M \sim \mathcal{Z}_n \quad (8)$$

Two algorithms, a_1 and a_2 , are compared in the experiments, each yielding a single fitted model per learning sample with a maximum of three inner knots, $k_{max} = 3$:

- a_1 fits $2, \dots, k_{max}$ models to the given learning sample by using the standard equal-sized quantiles approach and selects the model yielding the lowest BIC score.
- a_2 uses the knot selection process presented in this paper and selects the model with an inner knot count in the range $k = 0, \dots, k_{max}$ that yields the lowest BIC score.

By pairwise applying a_1 and a_2 to each of the m learning samples, we get the fitted models:

$$a_{im} = a_i(\cdot | \mathcal{L}^m), i = 1, 2 \quad (9)$$

The fitting procedures for the algorithms are deterministic, meaning they do not depend on random starting values or hyperparameters outside the learning samples. Also, the finished models, a_{im} , no longer depend on hyperparameters. Further, the models are themselves random variables depending on \mathcal{L}^m and have a distribution dependent on the data-generating process:

$$a_{im} \sim \mathcal{A}_i(\mathcal{Z}_n) \quad (10)$$

The model performances are measured with a scalar function, p , which also can be interpreted as a random variable with a distribution dependent on the data-generating process:

$$p_{im} = p(a_i, \mathcal{L}^m) = p(a_{im}) \sim \mathcal{P}_i = \mathcal{P}(\mathcal{Z}_n) \quad (11)$$

For each model-pair, a_{1m} and a_{2m} , fitted per learning sample, \mathcal{L}^m , we draw $t = 2,000$ observations from the same data-generating process as the learning samples, $z = (f, x, y) \in \mathfrak{Z}$, where $\mathfrak{Z} \sim \mathcal{Z}_t$. However, because predictions for RCS models are most reliable between the boundary knots, we ensure that the x -values are within this range.

The performance per model, \hat{p}_{im} , is computed by approximating the expected loss between the ground truth, $f = \phi(x)$, and the predicted value, $\hat{y} = a_{im}(x)$:

$$\hat{p}_{im} = \hat{p}(a_i, \mathcal{L}^m) = \frac{1}{t} \sum_{z=(f,x,y) \in \mathfrak{Z}} L(f, \hat{y}) \quad (12)$$

, where L is the quadratic loss function:

$$L(f, \hat{y}) = (f - \hat{y})^2. \quad (13)$$

This gives us two random samples consisting of M approximated performance measure values from the distributions $\mathcal{P}_1(\mathcal{Z}_n)$ and $\mathcal{P}_2(\mathcal{Z}_n)$, one set for each algorithm. We now formulate the null hypothesis, where $\hat{\mathcal{P}}_i$ is the approximation of \mathcal{P}_i . The null hypothesis is rejected at a significance level of $\alpha = 0.05$.

$$H_0 : E(\hat{\mathcal{P}}_1(\mathcal{Z}_n)) = E(\hat{\mathcal{P}}_2(\mathcal{Z}_n)) \quad (14)$$

Because the models for a_1 and a_2 are fitted to the same learning sets, the natural experimental design is a paired K samples design, as described

in section 4 of Hothorn, T. et al. (Hothorn et al, 2005). The paired difference test t -statistic is used under the null hypothesis of equality of the performance measure distributions.

5.2 Comparing BIC scores

In addition to hypothesis testing the distributions of the estimated performance measure, we also report the differences in BIC scores between a_1 and a_2 , as well as the sample mean difference in knot counts. These are meant as descriptive, whereas the main hypothesis is on the performance measure as described in the previous subsection.

6 Experiments and results

For the application example experiments, we followed the method described in Section 5. Four different non-linear functions for true curve were defined, $F = \phi(X)$. A lognormal distribution was used as \mathcal{X} (Eq. 15) for the experiments in Section 6.2, 6.3, and 6.4. For the experiment in Section 6.5, the distribution was uniform (Eq. 16). The error component distribution, \mathcal{E} , was homoscedastic and normal (Eq. 17) for all experiments. The values for the true curve, F , and error component, E , were added to obtain the values for Y (Eq. 18). 0.05 and 0.95 percentiles were used as lower and upper boundaries for the X observations for the samples. For the $t = 2,000$ observations, $z = (f, x, y) \in \mathfrak{Z}$, used to estimate the performance measure, \hat{p}_{im} , all values were between these boundaries. The reason is that RCS models are not reliable outside the interval of the predictor variable observations used to fit a model.

$$X \sim \text{Lognormal}(0.5, 0.35) \quad (15)$$

$$X \sim U(0, 3.5). \quad (16)$$

$$E \sim N(0, 0.1) \quad (17)$$

$$Y = F + E \quad (18)$$

6.1 Result report structure

The results from the two-tailed paired sample t -tests for the differences in estimated performance measures and BIC scores for the four experiments

are reported together in Table 1 in Section 6.6. The functions representing the true curves and figures illustrating the results are described in four subsections following the same structure (See Section 6.2 6.3, 6.4, and 6.5):

1. The experiment's ground truth curve function, ϕ_i , is briefly described and the formula defined. See Equation 19, 20, 21, 22.
2. The function description is followed by a plot showing an example of the resulting true and fitted curves for a single sample. The BIC score for the fitted curves are included in the plot's legend. See Fig. 5, 8, 11, and 14.
3. A figure showing the distributions of the estimated performance measure, \hat{p}_{im} , for a_1 and a_2 as boxplots within violin plots. See Fig. 6, 9, 12, and 15.
4. A figure showing the distributions of the BIC scores for a_1 and a_2 as boxplots within violin plots. See Fig. 7, 10, 13, and 16.

6.2 Logistic

The first curve function, Equation 19, is a logistic function. Such functions are seen in many fields, including sigmoid activation functions for artificial neural networks (Hastie et al, 2009, p. 392) and logistic regression (Kleinbaum, 2014, p. 682). However, logistic functions are also encountered in the context of non-linear relationships between predictor and response variables, where population growth (Campbell et al, 2007, p. 390) is one example.

Fig. 5 shows the function's curve and the curves approximated by a_1 and a_2 for an example sample of 250 observations. Fig. 6 shows the distributions of the estimated performance measure for a_1 and a_2 as box plots within violin plots. Fig. 7 shows a similar plot but for BIC scores.

$$\phi_1(X) = (1 + e^{-12X+27})^{-1} \quad (19)$$

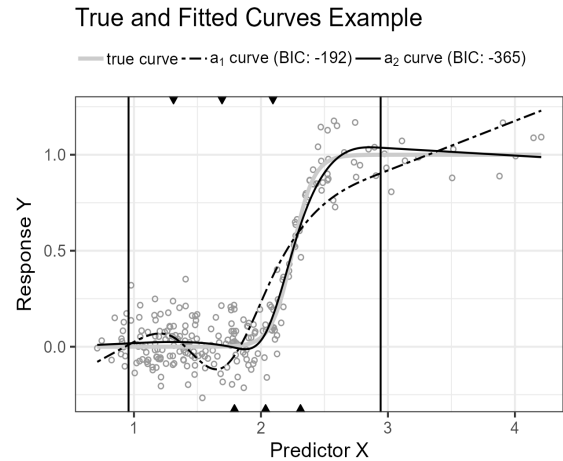


Fig. 5 The figure shows the logistic function's curve together with the curves approximated by a_1 and a_2 for an example sample of 250 observations. The vertical lines are the outer 0.05 and 0.95 percentile boundaries. The downwards-pointing triangles along the top horizontal axis mark the inner knot locations for a_1 , and the upwards-pointing triangles along the bottom axis are the knot locations for a_2 .

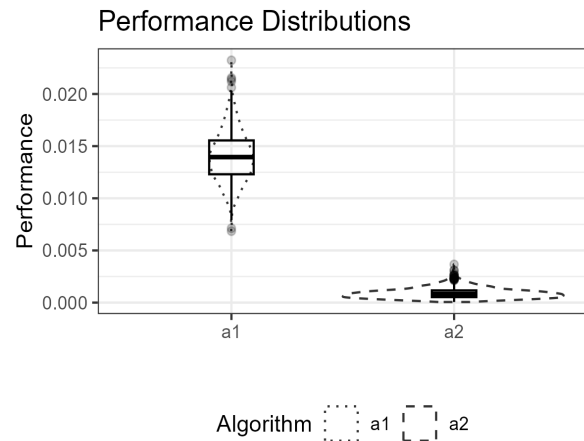


Fig. 6 The figure shows the distributions for $\hat{p}(a_1, \mathcal{L}^m)$ and $\hat{p}(a_2, \mathcal{L}^m)$ for the logistic function as box plots within violin plots.

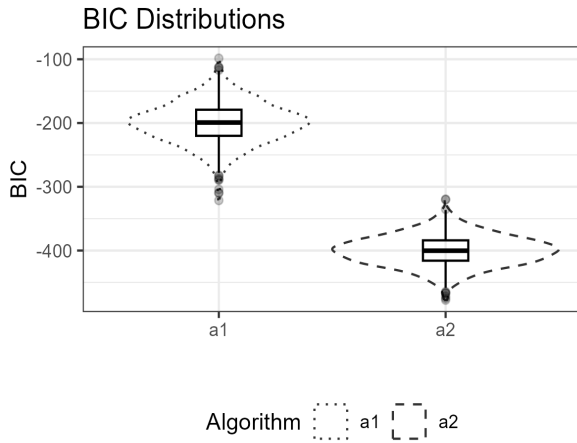


Fig. 7 The figure shows BIC distributions for the logistic function as box plots in violin plots.

6.3 Runge

The second function, Equation 20, is a Runge function. Such functions are commonly used to demonstrate Runge’s phenomenon where fitting high-order polynomials by equidistant interpolation points results in oscillation at the endpoints and failure to converge (Dahlquist and Björck, 1974, p. 101). Here, the curve is translated so that the central peak is at $X = \frac{8}{3}$. We do not use high-order polynomials, but lower-order fractional polynomials also perform worse than splines for this function.

Fig. 8 shows the function’s curve and the curves approximated by a_1 and a_2 for an example sample of 250 observations. Fig. 9 shows the distributions of the estimated performance measure for a_1 and a_2 as box plots within violin plots. Fig. 10 shows a similar plot but for BIC scores.

$$\phi_2(X) = \frac{1}{1 + (\frac{3}{2}X - 4)^2} \quad (20)$$

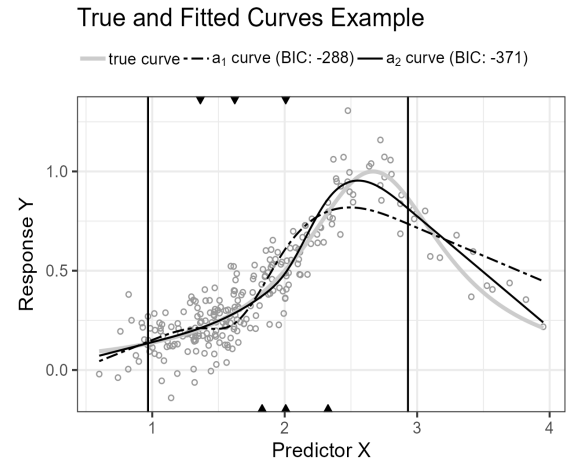


Fig. 8 The figure shows the Runge function’s curve together with the curves approximated by a_1 and a_2 for an example sample of 250 observations. The vertical lines are the outer 0.05 and 0.95 percentile boundaries. The downwards-pointing triangles along the top horizontal axis mark the inner knot locations for a_1 , and the upwards-pointing triangles along the bottom axis are the knot locations for a_2 .

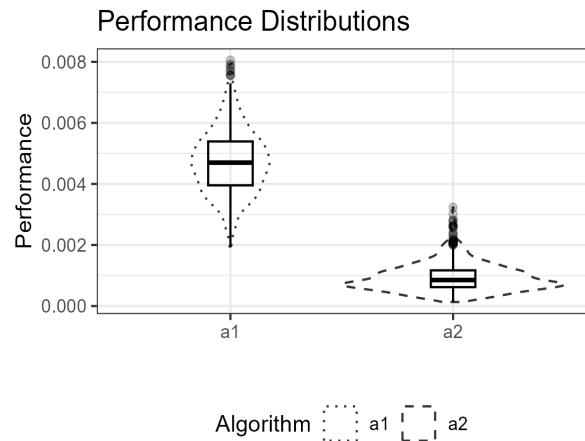


Fig. 9 The figure shows the distributions for $\hat{p}(a_1, \mathcal{L}^m)$ and $\hat{p}(a_2, \mathcal{L}^m)$ for the Runge function as box plots within violin plots.

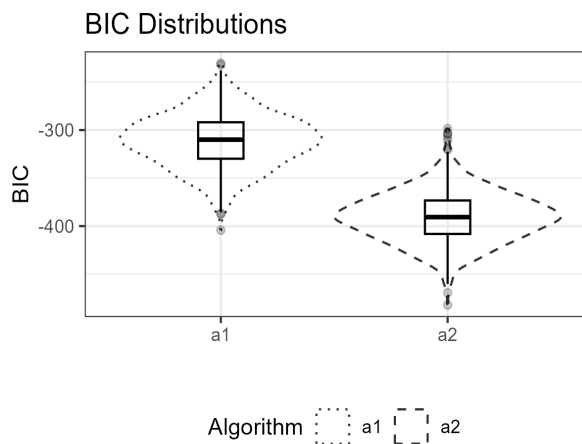


Fig. 10 The figure shows BIC distributions for the Runge function as box plots in violin plots.

6.4 Trigonometric

The third function, Equation 21, is a cosine function. The functions turning points require an economic placement of the three available inner knots.

Fig. 11 shows the function's curve and the curves approximated by a_1 and a_2 for an example sample of 250 observations. Fig. 12 shows the distributions of the estimated performance measure for a_1 and a_2 as box plots within violin plots. Fig. 13 shows a similar plot but for BIC scores.

$$\phi_3(X) = \cos\left(\frac{3\pi}{2}X\right) \quad (21)$$

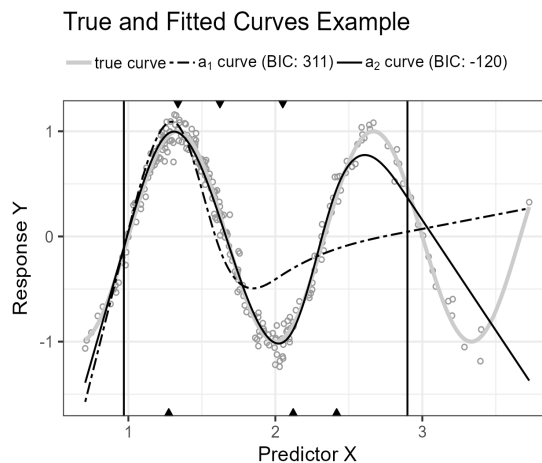


Fig. 11 The figure shows the trigonometric function's curve together with the curves approximated by a_1 and a_2 for an example sample of 250 observations. The vertical lines are the outer 0.05 and 0.95 percentile boundaries. The downwards-pointing triangles along the top horizontal axis mark the inner knot locations for a_1 , and the upwards-pointing triangles along the bottom axis are the knot locations for a_2 .

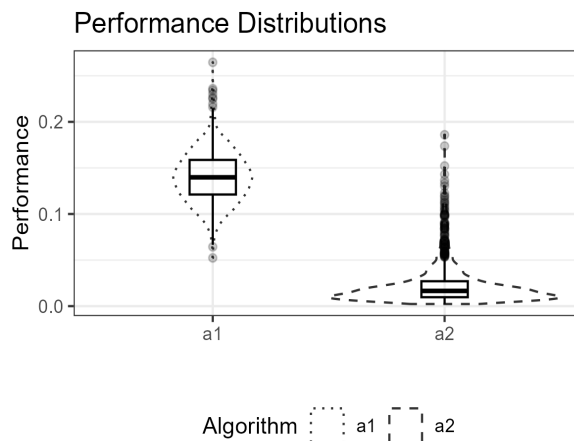


Fig. 12 The figure shows the distributions for $\hat{p}(a_1, \mathcal{L}^m)$ and $\hat{p}(a_2, \mathcal{L}^m)$ for the trigonometric function as box plots within violin plots.

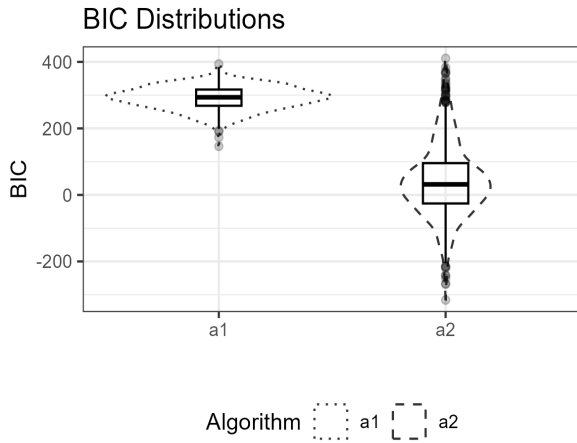


Fig. 13 The figure shows BIC distributions for the trigonometric function as box plots in violin plots.

6.5 Gaussians

The last curve function, Equation 22, is the sum of two Gaussian functions reminiscent of a bimodal distribution with two normal distributions, $\frac{3}{2}\mathcal{N}(1, 0.5)$ and $\mathcal{N}(2.75, 0.5)$. As the only experiment, X has a uniform distribution, $X \sim U(0, 3.5)$, in contrast to the other three experiments that use a lognormal distribution.

Fig. 14 shows the function's curve and the curves approximated by a_1 and a_2 for an example sample of 250 observations. Fig. 15 shows the distributions of the estimated performance measure for a_1 and a_2 as box plots within violin plots. Fig. 16 shows a similar plot but for BIC scores.

$$\phi_4(X) = \frac{3}{2}\gamma(X, 1, \frac{1}{2}) + \gamma(X, \frac{11}{4}, \frac{1}{2}) \quad (22)$$

, where $\gamma(X, a, b)$ is the Gaussian function,

$$\gamma(X, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(X - \mu)^2}{\sigma^2}\right). \quad (23)$$

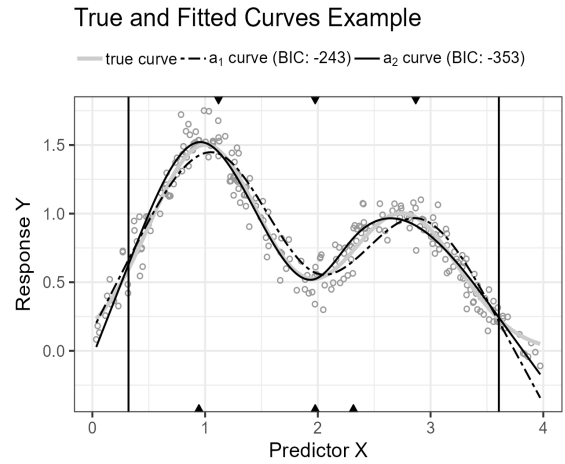


Fig. 14 The figure shows the Gaussians function's curve together with the curves approximated by a_1 and a_2 for an example sample of 250 observations. The vertical lines are the outer 0.05 and 0.95 percentile boundaries. The downwards-pointing triangles along the top horizontal axis mark the inner knot locations for a_1 , and the upwards-pointing triangles along the bottom axis are the knot locations for a_2 .

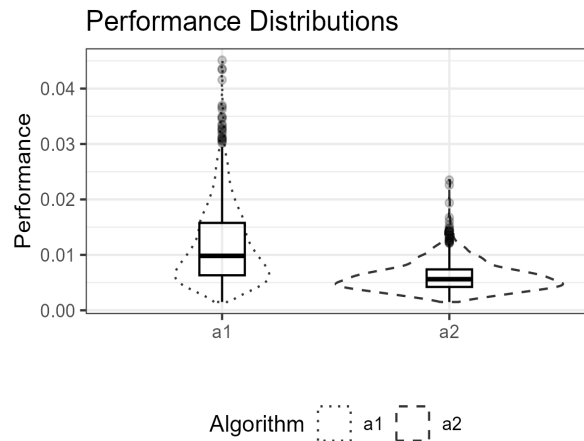


Fig. 15 The figure shows the distributions for $\hat{p}(a_1, \mathcal{L}^m)$ and $\hat{p}(a_2, \mathcal{L}^m)$ for the Gaussians function as box plots within violin plots.

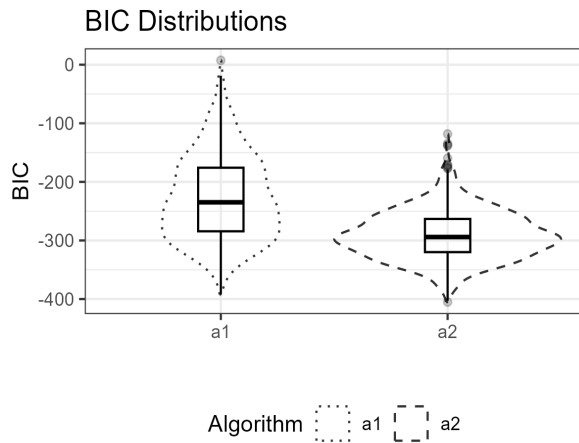


Fig. 16 The figure shows BIC distributions for the Gaussians function as box plots in violin plots.

6.6 Results table

Table 1 shows the results for the four experiments described in Section 6.2, 6.3, 6.4, and 6.5. The columns are:

1. The function representing the true curve in the experiment
2. The 95% confidence interval for the expected difference in estimated performance measure for a_1 and a_2
3. The performance measure sample mean for a_1
4. The performance measure sample mean for a_2
5. The 95% confidence interval for the expected difference in BIC scores for a_1 and a_2
6. The sample mean for the number of knots for a_1 minus the number of knots for a_2

7 Discussion

The implementation of the knot selection algorithm presented in this paper is part of an R-package, `knutar`. The package-function `choose_model` uses the algorithm but only selects the model produced if it scores better than the models from fractional polynomial regression and the standard knot selection process for RCS. Thus, we do not intend the presented process to replace the standard knot selection process but as an option in addition to it. The results can often be similar, so the processes agree. However, as the experiments show, there are cases where the models produced by our knot selection process perform significantly better.

For all four experiments, Section 6.2–6.5, the null hypothesis, Equation 14, was rejected. We do not report the exact p values in the results table. The reason is that we can, in principle, generate infinitely many artificial test observations, and the p value will reach zero in the limiting case when the two distributions are different. We can detect performance differences with high power. However, as discussed by Hothorn et al. (Hothorn et al, 2005, p. 697), "one should always keep in mind that statistical significance does not imply a practically relevant discrepancy and therefore the amount of the difference should be inspected by confidence intervals and judged in the light of analytic expertise." Instead, we can turn to the presented confidence intervals to inspect the amount of difference. Especially the expected difference in BIC scores for a_1 and a_2 reported in Table 1 suggests that our knot selection process achieves a clear improvement compared to the standard process for the example experiments.

Predictor variables usually have non-uniform distribution for real data. When equal-sized quantiles separate knots, more knots are located in dense regions than in sparser regions of the predictor variable. Suppose the number of knots is relatively low compared to the non-linear curve shape for the relationship between the predictor and response variables. Placing inner knots by a regular sequence of quantiles may lead to too few knots and underperformance in the sparser regions, i.e., discrepancy due to approximation, which is the case for the right-side regions of Figures 5, 8, and 11. In these regions, our knot selection process shows a better adaptation to the true curve. In Fig. 11, we see that there are more critical points than in Fig. 5 and 8. In such cases, the selected locations for the few available knots become more crucial to the resulting goodness of fit.

Conversely, the standard knot selection process can lead to overfitting in denser regions. For example, in Figure 5 and 8, we see that the curve for a_1 is wiggly compared to the true curve, indicating an overfitted a_1 curve.

The experiment in section 6.5 uses a uniform distribution for X . Therefore, the density of the predictor variable observations does not systematically differ in any particular region of the sample for the $M = 1,000$ learning samples. However, the ground truth curve for the experiment has a

Table 1 Experiment results

| <i>Curve</i> | $E(\hat{\mathcal{P}}_1) - E(\hat{\mathcal{P}}_2)$ CI | $\bar{\mathcal{P}}_1$ | $\bar{\mathcal{P}}_2$ | $E(BIC(a_{1m}) - BIC(a_{2m}))$ CI | \bar{d}_k |
|---------------|--|-----------------------|-----------------------|-----------------------------------|-------------|
| Logistic | [1.295e-02, 1.326e-02] | 1.401e-02 | 9.086e-04 | [198, 203] | 0.0 |
| Runge | [3.721e-03, 3.848e-03] | 4.724e-03 | 9.391e-04 | [78, 81] | 0.2 |
| Trigonometric | [1.144e-01, 1.186e-01] | 1.401e-01 | 2.354e-02 | [244, 258] | 0.0 |
| Gaussians | [5.187e-03, 6.105e-03] | 1.175e-02 | 6.102e-03 | [59, 69] | 0.0 |

bimodal shape with turning points that do not necessarily align well with a few knots distanced by equal-sized quantiles. Here, the presented selection process has greater flexibility in placing knots and can better fit a regression spline using the same number of knots.

RCS models are often unreliable in the tails, i.e., before the first boundary knot and after the last. For this reason, we have only assessed the performance of predictions given predictor variable values between the boundary knots of the fitted models, as described in Section 5.1, which prevents outliers in the tails from causing extreme squared prediction error values that distort the performance measure. The same is not the case for BIC scores. In our context, BIC scores are used for model selection, including models not stemming from RCS regression. Thus, we choose to compute BIC scores without customizing specifically for RCS regression models.

Lastly, we briefly discuss model selection bias (Zucchini, 2000, pp. 58-60). The knot selection process presented in this paper uses a backward method that assesses many models. Nevertheless, in the end, only the best final models from the knot selection processes are compared in the experiments. It could be that the BIC score systematically is more favorable for one selection process than another for unknown reasons. However, in the simulation experiments, the primary performance measure and hypothesis testing are not based on BIC but on the ground truth and predictions for $t = 2,000$ previously unseen observations for each $m = 1,000$ finished models per knot selection process. Therefore, the estimated performance measures should not be affected by selection bias.

8 Related work

Several advanced methods exist for spline regression. A prominent example is penalized B-splines (P-spline) (Eilers and Marx, 2021; Eilers et al, 2015), where smoothing splines (Hastie et al, 2009, pp. 151-153) is a specialized case. It is a flexible framework where splines are built from the sum of basis curves scaled by coefficients, most commonly a high number of equally distanced B-splines. The P-spline method prefers an abundance of knots and control overfitting by a roughness penalty (regularization) that smoothes or dampens the wiggleness of the curve instead of reducing the complexity of the model by removing knots. By setting coefficients to zero, certain B-splines in the mixture can, in effect, be removed. Although superficially similar to RCS regression, B-splines and P-splines are different methods from RCS regression. The knot selection presented in this paper concerns the latter.

Other techniques adaptively place knots or choose spline basis functions. Typically, these advanced methods produce relatively complex models or target slightly different problems, such as hybrid adaptive splines (Luo and Wahba, 1997) for when there is an underlying function that is spatially inhomogeneous in its degree of complexity. In comparison, restricted cubic splines can easily be used to include non-linear relationships in a wide variety of models (Buis, 2009).

9 Conclusion

We have presented a knot selection process and greedy state space search algorithm for RCS regression and implemented it as part of an open-source R-package, `knutar`. The example simulation experiments show lower prediction errors and improved goodness of fit compared to placing an equivalent number of inner knots by a regular

sequence of quantiles. The presented knot selection process can be used as an alternative to the standard process when the curve approximation is challenging due to several critical points, regions where the predictor variable's observations are sparse, or both. It can also reduce overfitting in the more densely distributed regions of the predictor variable observations.

10 Data Availability

The datasets generated and analyzed during the current study are available in the `knutar-experiments` repository, <https://github.com/jo-inge-arnes/knutar-experiments>

11 Declarations

11.1 Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

- Ambler, G., Benner, A.: `mfp`: multivariable fractional polynomials. <https://CRAN.R-project.org/package=mfp> (2022). Accessed 3 March 2023
- Buis, M.L.: Using and interpreting restricted cubic splines. 7th German Stata Users Group Meeting, Bonn, Germany. <http://www.maartenbuis.nl/presentations/bonn09.pdf> (2009). Accessed 3 March 2023
- Campbell, N.A., Reece, J.B., Simon, E.J.: *Essential Biology*, 3rd edn. Pearson Benjamin Cummings, San Francisco, CA (2007)
- Chatfield, C.: Model uncertainty, data mining and statistical inference. *J. R. Stat. Soc. Ser. A Stat. Soc.* (1995) <https://doi.org/10.2307/2983440>
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., et al.: *Introduction to Algorithms*, 4th edn. MIT, Cambridge, MA (2022)
- Dahlquist, G., Björck, A.: *Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ (1974)
- Eilers, P.H.C., Marx, B.D., Durban, M.: Twenty years of p-splines. *SORT*. 39, 149–186 (2015)
- Eilers, P.H.C., Marx, B.D.: *Practical Smoothing: The Joys of P-splines*. Cambridge University, Cambridge, England (2021)
- Gauthier, J., Wu, Q.V., Gooley, T.A.: Cubic splines to model relationships between continuous variables and outcomes: a guide for clinicians. *Bone Marrow Transplant.* (2020). <https://doi.org/10.1038/s41409-019-0679-x>
- Harrell, F.E, Jr.: *Regression Modeling Strategies*, 2nd edn. Springer, Cham, Switzerland (2015)
- Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*, 2nd edn. Springer, New York, NY (2009)
- Hothorn, T., Leisch, F., Zeileis, A., et al.: The design and analysis of benchmark experiments. *J. Comput. Graph. Stat.* (2005). <https://doi.org/10.1198/106186005X59630>
- Kleinbaum, D.G.: *Applied Regression Analysis and Other Multivariable Methods*, 5th edn. Cengage, Boston, MA (2014)
- Luo, Z., Wahba, G.: Hybrid adaptive splines. *J. Am. Stat. Assoc.* (1997) <https://doi.org/10.1080/01621459.1997.10473607>
- Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., et al.: A review of spline function procedures in r. *BMC Med. Res. Methodol.* (2019). <https://doi.org/10.1186/s12874-019-0666-3>
- R Core Team: *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/> (2022). Accessed 3 March 2023
- Royston, P., Altman, D.G.: Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling. *J. R. Stat. Soc. Ser. C Appl. Stat.* (1994). <https://doi.org/10.2307/2986270>
- Russell, S., Norvig, P.: *Artificial Intelligence : A Modern Approach*, 3rd edn. Pearson, Boston,

MA (2010)

Sauerbrei, W., Abrahamowicz, M., Altman, D.G., et al.: Strengthening analytical thinking for observational studies: the stratos initiative. *Stat. Med.* (2014). <https://doi.org/10.1002/sim.6265>

Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* (1978). <https://doi.org/10.1214/aos/1176344136>

Steyerberg, E.W.: *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*, 2nd edn. Springer, Cham, Switzerland (2019)

Stone, C.J.: Comment: Generalized additive models. *Stat. Sci.* (1986). <https://doi.org/10.1214/ss/1177013607>

Zucchini, W.: An introduction to model selection. *J. Math. Psychol.* (2000). <https://doi.org/10.1006/jmps.1999.1276>

Paper IV

Schema-Based Priming of Large Language Model for Data Object Validation Compliance

Jo Inge Arnes and Alexander Horsch

Under revision. Preprint DOI: [10.2139/ssrn.4453361](https://doi.org/10.2139/ssrn.4453361)

Schema-Based Priming of Large Language Model for Data Object Validation Compliance

Jo Inge Arnes^{a,*}, Alexander Horsch^a

^a*Department of Computer Science, UiT The Arctic University of Norway, Hansine Hansens veg 18, Tromsø, 9019, Norway*

Abstract

General-purpose large language models (LLMs), such as OpenAI's GPT models, can be integrated into applications to enable natural-language user interfaces. However, making a general LLM comply with validation rules for complex data structures used by in-house systems can be challenging. Two common ways to customize LLMs are fine-tuning and prompt engineering. We propose a prompt-based approach enabling general LLMs to stepwise and correctly mutate structured data objects complying with validation rules without specific training data. Our approach utilizes well-known schema standards to define the rule set and dynamically generates system messages or initial prompts to instruct the LLM. We implement a prototype with a client application handling communication between the LLM, a user, and a domain-specific backend system. The proposed approach has broad applications for scenarios where LLMs must manage data objects having rules for valid structure and values. This is, for instance, the case when interacting with external systems or assisting users in executing tasks with predefined steps or configuration options.

Keywords: natural-language user interface, prompt engineering, schema validation

1. Introduction

Large language models (LLMs) have undoubtedly caught the general public's attention with OpenAI's release of GPT-4 [1, 2], easily accessible through a chat-like service on the web, ChatGPT [3]. LLMs commonly utilize deep-learning [4] architectures, such as the Transformer [5] architecture and generative pre-training [6], designed to understand and generate human-like text by being trained on colossal general datasets. The models can perform various tasks, including text summarization, translation, question-answering, and conversational AI. LLMs achieve a broad perceived understanding of human

*Corresponding author

Email address: jo.i.arnes@uit.no (Jo Inge Arnes)

language by learning patterns and structures from the training data. This enables them to generate contextually appropriate responses and to accurately complete various language-related tasks.

However, the most popular LLMs are general-purpose and usually do not know the specifics of in-house systems or work processes that may require complex but clearly defined rules, structures, and configurations, for example, regarding workflows, processes, or system states. Consequently, making the LLM comply with the strict requirements can be challenging, mainly because such language models often generate convincing but fictitious and erroneous responses when the answer is basically unknown. In [2, pp. 1-2], OpenAI states the limitations of GPT-4 and earlier models and warns that the models can be unreliable and suffer from "hallucinations." They also have a limited context window and do not learn from experience. Therefore, particular care should be taken when using the outputs where reliability is essential. Further, they identify the careful study of these challenges as an important area of research.

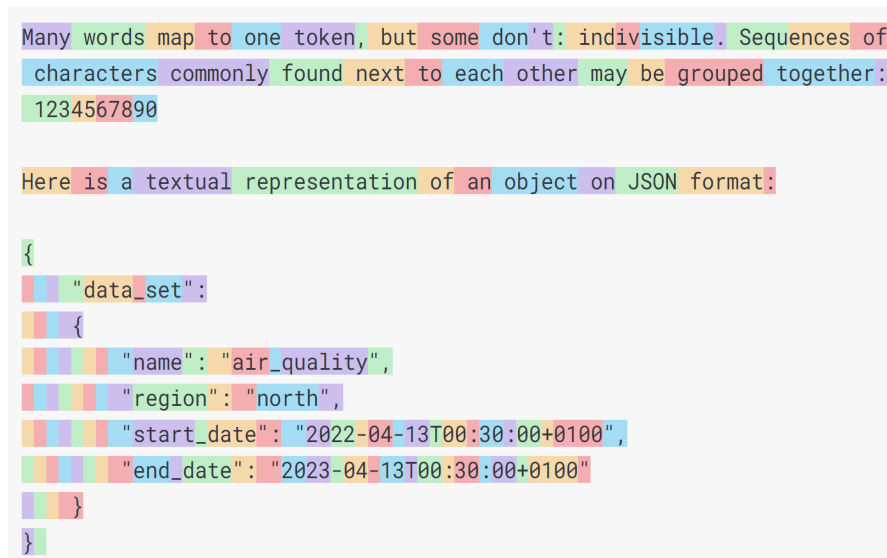
Thus, using general LLMs to enable natural-language user-interfaces in conjunction with in-house systems can be difficult. This is unfortunate because the models otherwise have plenty of desirable properties. Here we propose an approach to instructing the LLM to comprehend and enforce a strict set of rules by utilizing machine-interpretable, text-based standards for defining and validating results.

One common way of customizing a general LLM is fine-tuning [7]. Fine-tuning involves training a general model on a specific task or domain by providing a training dataset with relevant examples of prompts and completions specific to the task or domain. The resulting fine-tuned model can be seen as an adjusted version of the original model, which can be stored separately. Some commercial companies offering the most capable LLMs as an online service, such as OpenAI, allow fine-tuning of their models for a fee based on the chosen base model and the data size measured in the number of tokens. One thousand tokens equal approximately 750 words [8], but that number may vary. An alternative is to download and fine-tune models on-premise, utilizing either central processing units (CPUs) or graphical processing units (GPUs). There are many open and free-to-use LLMs available, for example, on Huggingface (<https://huggingface.co>).

Many open models are fine-tuned versions of Meta AI's GPL-licenced LLaMA [9] models, for example, Vicuna-13B [10]. The LLaMA models have from 7 to 65 billion parameters. GPT-J [11] is another open model. It is Apache licensed and has 6 billion parameters. GPT-J has been used as a base model for, e.g., Databricks' Dolly 1.0 [12], and NomicAI's GPT4All-J [13]. GPT4All-J can be downloaded and run locally on a PC. Databricks' Dolly 2.0 [14] is based on another EleutherAI model, Pythia [15], having 12 billion parameters. In comparison, GPT-3 has 175 billion parameters. The number of parameters for GPT-4 has not been officially disclosed, but news media has reported around a trillion parameters [16]. Because deploying independent instances of fine-tuned models can become prohibitively expensive due to the high number of parameters, techniques exist that can reduce the number of trained parameters.

For example, Low-Rank Adaptation (LoRA) reduces the number of parameters trained by 10,000 times and the GPU memory requirement by three times [17].

Another common way of customization is prompt engineering [18], which involves designing input prompts that can guide the model to produce responses more aligned with the given needs without modifying the underlying model. Priming the model with prompts can be more flexible and less time-consuming than fine-tuning because prompts are given as input on-the-fly to the model and may vary from session to session. Fine-tuning, in contrast, requires a training dataset of examples having sufficient quality and variation, and the model’s knowledge does not change dynamically after the training. A drawback of priming the model using prompting is that it may be less exact. Another drawback is that the models have a limited context window, meaning they can only handle a limited number of word tokens as input for producing a text completion. Also, the underlying model does not change due to prompts and completions. Each completion request includes a log of previous prompts and completions must fit within the model’s token limit. Fig. 1 shows how text can be split into tokens by the LLM, where the tokens are the units processed by the LLM rather than words.



```
Many words map to one token, but some don't: indivisible. Sequences of
characters commonly found next to each other may be grouped together:
1234567890

Here is a textual representation of an object on JSON format:

{
  "data_set":
  {
    "name": "air_quality",
    "region": "north",
    "start_date": "2022-04-13T00:30:00+0100",
    "end_date": "2023-04-13T00:30:00+0100"
  }
}
```

Figure 1: The figure shows tokenization as performed by OpenAI’s GPT-3 (<https://platform.openai.com/tokenizer>). The characters have been given different background colors to indicate their grouping into tokens. The text is 404 characters long, split into 163 tokens.

We propose a prompt-based procedure to enable a general LLM to stepwise mutate a given structured data object into a valid data object according to strict rules governing the structure, possible and required fields, and allowed values without this being represented in the LLM’s training data. Our approach defines the rule set for the structured data object by utilizing well-known schema standards, which can also contain meta-data with human-readable descriptions.

Such a schema can, for example, be dynamically generated as part of a REST API [19, pp. 31-32] for a system used by an organization. Our proposed solution will then be able to generate a system message or initial prompt as a first instruction to the LLM that also includes the schema defining the validation rules for the object structure. The structured data object, which is to be mutated step by step by the LLM, can more concretely describe a job, configuration, or state object that the external system can receive and execute. Such objects can even be used for defining a program in a domain-specific language (DSL) [20]. Currently, plugins [21] and built-in solutions for communicating with external systems are non-public for the most relevant LLM services, such as OpenAI’s ChatGPT. Instead, our implemented prototype has a client application that handles the communication between the LLM, a user, and a domain-specific backend system.

The approach proposed in this paper can have many areas of application when LLMs must handle a valid state object on a specific structure, communicate with external systems, or assist a user in the execution of tasks with fixed steps or configuration options.

The rest of the paper is organized as follows: First, we describe an example scenario to clarify the background of our approach in Section 2. Section 3 presents our approach’s overall architecture and design. In Section 4, we describe the implementation of the prototype. Section 7 presents experiments and results. The discussion is in Section 8 before concluding in 9.

2. User scenario

Before presenting our approach, we describe an example scenario used as a common thread throughout the rest of the paper. We do this so the reader can more readily understand the motivation behind the proposed approach. Still, it is essential to appreciate that this is merely one of many possible applications.

In the user scenario, a scientist wants to analyze air quality data to identify patterns and trends in air pollution and visualize the data. The scientist uses a system custom to an organization specializing in environmental conservation. The system has a library of available modules with functions usable as part of an analysis pipeline. A team of scientific programmers is responsible for the development, regularly publishing updates and new modules. The scientist must choose functions and parameter values to define an analysis pipeline that will be submitted to a backend scheduler and executed on a compute cluster. The scientist uses a client application with an integrated natural-language user interface to configure and submit the job. The job configuration is represented in a text-based format for structured data and must comply with specific rules to be valid. The job consists of three main steps:

1. Data preprocessing
2. Data analysis
3. Data visualization

For the first step, the scientist chooses a function to filter the data on a pollutant and a date range. For the second step, the scientist chooses trend analysis and sets parameters. Finally, for the third stage, the scientist selects a line chart to visualize the trends in pollutant concentration over time. Once finished configuring the job, the scientist submits the job via the backend’s REST API for scheduling and execution on a cluster. The cluster then preprocesses, analyzes, and generates the desired visualizations.

The payload submitted to the backend via its API is a text-based representation of a structured data object, which must comply with specific rules. For example, the job consists of three defined steps, each comprising a selection of module functions to be applied sequentially. The functions are tied to specific steps, so a function belonging to one step is not usable in another. The different functions can have unique required and optional parameters. All parameter values have a specified data type, and there can be restrictions on valid values. For user-friendliness, the developers have provided meta-data documenting the different functions and parameters in human-readable form.

The natural-language UI must understand and enforce these requirements, explain the rules, and help the user build a valid job object stepwise through a chat-like interaction. This is a suitable scenario to illustrate our approach.

3. Architecture and design

The component diagram in Fig. 2 shows the overall architecture of the proposed approach. Returning to the scenario described in Section 2, the scientist uses a client-side application to configure the analysis job, primarily through a chat-like interaction within the client’s user interface (UI). The UI corresponds to the architecture’s `view` component. The client application has a `conversation state` component to keep track of the chat log. The natural-language functionality is enabled by the LLM component. In the scenario, the scientist configures the job in a stepwise manner, and the `object state` component is responsible for managing the current job configuration object. A schema defines the job object’s structure and valid values and is retrieved from the `schemas` component, which is part of a backend service in our scenario. Lastly, when the scientist has finished configuring, the job object can be submitted to an `executor` component for execution. However, these components do not communicate directly with each other. Instead, a `controller` coordinates the communication. Note that the organization of the processes into client, backend, and LLM is a guideline, not a mandatory part of the architecture.

3.1. Process interaction

The sequence diagram in Fig. 3 shows how the processes interact. We have chosen a higher level of abstraction and do not show the individual components of the processes in the diagram.

In our scenario, the client application prepares the natural-language interface for its task before being used by the scientist. Therefore, the client’s `controller`

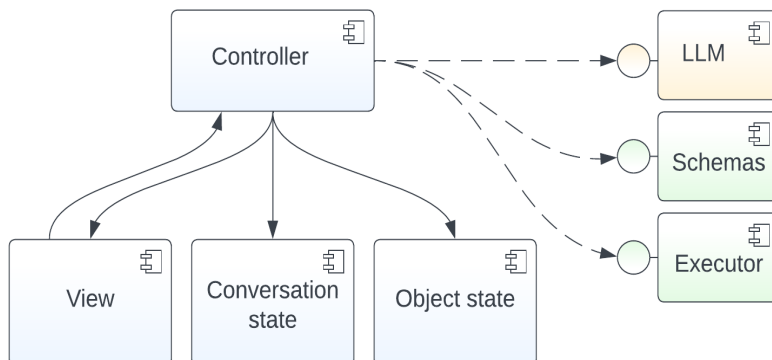


Figure 2: The component diagram illustrates the overall architecture. The components are shown as boxes, and the arrows indicate direct communication. The component at the origin of an arrow can call the component at the end. An arrow with a solid line means the calls are within the same process, whereas a dashed line indicates that the calls are inter-process. The different background colors group the components into processes. For example, in the scenario described in Section 2, the controller, view, conversation state, and object state are part of the client process, whereas the schemas and executor components are part of the backend, and an external service hosts the large language model (LLM).

component first obtains the schema by calling the backend’s `schemas` component API, which additionally returns an empty object. Then, internally to the client, the controller passes the empty object and the schema to the client’s `object state` component.

The client then prepares an initial prompt for priming the LLM, which includes a text representation of the schema and initial object. This prompt is a system message that will be hidden from the UI. The message also includes an instruction to greet the user, and the client’s controller makes an API call on the LLM service with the message as input and receives a completion. The controller passes the prompt and completion to the `conversation state` component and triggers the `view` to update the UI’s chat interface.

Here, the sequence diagram shows additional details. The system message also has a part instructing the LLM to include a text representation of the current or updated version of the object as part of the returned completion. The client extracts the text representation of the object from the completion, validates it according to the schema, deserialize it if necessary, and stores it in the `object state`. If the LLM has returned an invalid object, the client can optionally ask the LLM for a corrected version. Further, the client will hide the text representation of the object from the user by removing it from the completion. It may also do additional formatting. Thus, the `conversion log` keeps both a user-friendly and raw chat log.

The natural-language interface is now ready, and the scientist can start chat-

ting to configure the job. The technical staff, which develops the modules, has provided human-readable descriptions of the modules and parameters. This meta-data is included in the schema. In addition to the formal definitions of the object structure, the meta-data enables the natural-language interface to describe and help the scientist understand the options.

The sequence diagram illustrates that the chat interaction continues as a loop with prompts and completions, stepwise mutating the object state until the scientist is finished configuring and ready to submit to the job.

Recall that the LLM does not automatically remember previous prompts and completions in the conversation. Thus, the client passes the complete conversation and the system message for each completion request. However, the LLM accepts only a limited number of tokens. In our approach, the client excludes older messages from the request to prevent exceeding the token limit. As part of this truncation of the conversation, the system message is updated to contain the current object state instead of the empty default object. Optionally, the LLM can be asked to create a brief summary of the conversation so far, which can be included in the system message to provide the LLM with some memory of the excluded parts of the conversation.

Finally, when the scientist finishes the configuration, the completed job object, kept by the `object state` component, can be submitted to the backend system for job scheduling and execution on a cluster.

3.2. System message and completion request text

We now describe the system message and text sent to the LLM for completion in more detail. The system message is the prompt that primes the LLM for its custom task and is sent as the first text of all completion requests to the LLM. Because LLMs have a context window, i.e., a limited number of tokens in the text for which it will generate a completion, the system message must be short enough also to fit at least a user prompt and its completion because the token limit includes the completion.

However, it is desirable that at least some of the latest prompt-completion pairs are provided because operations may be split over several steps. For example, suppose the scientist in our scenario tells the natural-language interface to include and preprocess air quality data for a region. In that case, the natural-language interface may request a date range before creating an updated version of the job configuration object. If the scientist provides a date range in the chat, but the completion request to the LLM does not contain the preceding context, it will not understand what to do with the date range in isolation. Nevertheless, prompts and completions far back in the conversation history are not likely needed because they concern no longer relevant operations. If it is relevant to include more of the previous context from the chat history, we propose a method where the LLM is asked to summarize the conversation and have this as part of the system message. A previous summary can also be included when making a new one to replace it.

In our approach, the system message has sections with different purposes, as illustrated in Fig. 4. The first section is a short description setting the context

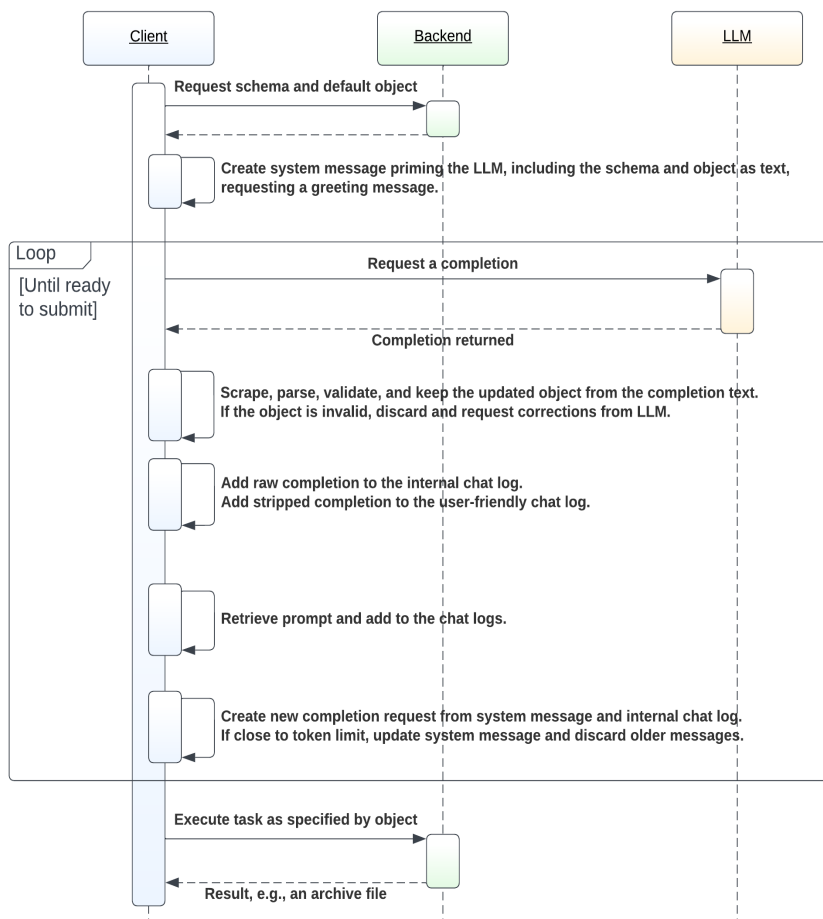


Figure 3: The sequence diagram illustrates a sequence of calls performed by the client in our architecture.

and the role of the natural-language interface. Next, the text explains that the task is to build an object structure and that the object must be valid according to a schema, followed by the text representation of the schema. The schema section will typically be the longest of the sections in the system message, depending on the complexity of the object structure, its validation rules, and meta-data. A text representation of the object itself follows it. After the schema and object sections, there are instructions restricting how the natural-language interface is allowed to help the user, and it also specifies details of how to respond. A crucial part of this section is that the text instructs the LLM to include the current object structure in the completion. This feature makes it possible for the client to extract the updated object structure and store it in the **object state**.



Figure 4: The figure illustrates the sections of the system message. The different box sizes do not represent exact ratios, but a larger box does indicate a generally larger section.

Finally, the text asks the LLM to respond with a welcome message. Alternatively, it may include a conversation summary thus far or other auxiliary information due to the truncation of the chat log. In the latter case, the object section will contain an updated version of the object structure.

4. Implementation

This section describes the relevant parts of the implemented prototype, which will be available at <https://github.com/jo-inge-arnes/schema-priming> on publication of the paper. We provide details and examples tied to topics described in Section 3. The prototype consists of a client application implemented as a single-page application (SPA) using VUE 3 (<https://vuejs.org/>), which runs in a browser. We have also implemented a backend using Python and Flask (<https://flask.palletsprojects.com>). However, the paper focuses on LLMs and schemas, making the backend implementation less critical. Fig. 5 shows a screenshot of the client. On the right-hand side is a chat-like interface. On the left-hand side, the current object state is shown. For the GPT models, the client calls the LLM service programmatically by making asynchronous HTTP requests to OpenAI’s API [22]. For local LLMs, we instead use a library for the given programming language providing functions for calling the LLM locally.

4.1. Object and schema representation

LLMs are text-based, and the object structure is serialized to a text format. Three major text-based standards for object representation are XML, JSON,

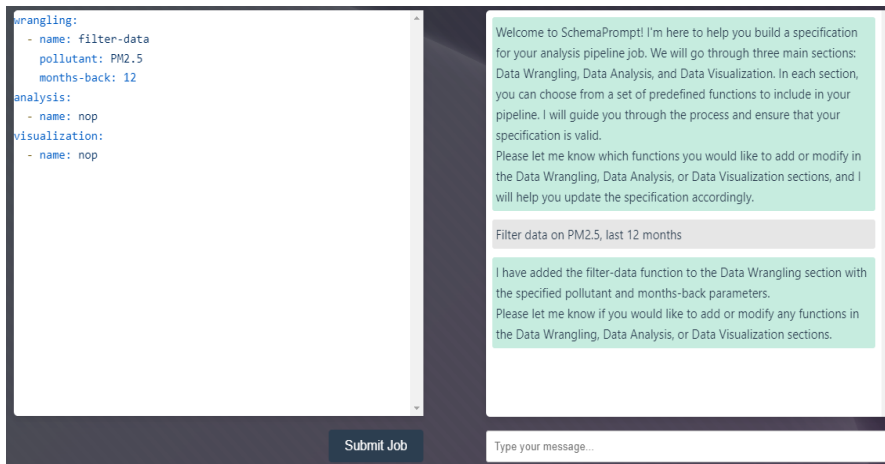


Figure 5: The figure shows a screenshot of the client application for the prototype. The chat interface is on the right side of the client window, whereas the YAML representation of the current object state is displayed on the left.

and YAML. XML is the oldest and most verbose of the three, whereas YAML is the youngest and least verbose. All are well-suited for data exchange and storage, configuration files, web-based APIs and services, and document metadata. Further, schema standards for object validation exist for all of them. Coverage of the topic is found in [19, pp. 151-176].

Because of token limits, we want to use a succinct standard for the text-representation objects and schemas. YAML is the less verbose of the three standards mentioned above and is used for representing the object structure. For the choice of schema standard, both XML schema and JSON schema need relatively many tokens. Further, YAML Schema is also relatively verbose because it has a JSON format. Instead of inventing a custom schema, we prefer a standard already implemented by programming language tools and libraries. It is also crucial that the standard includes more advanced control of what is allowed and what is not. The prototype in this paper uses the Kwalify (<https://metacpan.org/pod/Kwalify>) schema standard and, more specifically, the Python-based implementation Pykwalify (<https://pykwalify.readthedocs.io>).

5. Example object and schema

In this section, we first list the YAML representation of an object representing an initial empty job in our analysis pipeline configuration scenario in Listing 1. Next, the Kwalify-schema is listed in Listing 2, which is in YAML format.

Listing 1: The initial default object represented as YAML. It is 34 tokens long using the GPT-3 tokenizer.

```

wrangling:

```

```

-
  name: nop
analysis:
-
  name: nop
visualization:
-
  name: nop

```

Listing 2: The schema on Kwalify format. It is 1,101 tokens long using the GPT-3 tokenizer.

```

map:
  wrangling:
    desc: "Data Wrangling"
    required: True
    seq:
      - map:
          name:
            type: str
            pattern: filter-data
            required: True
            desc: "Function for filterings data, e.g., pollutant = 'PM2.5' and
              months-back = 12 to filter the dataset to only include PM2.5
              data from the last year."
          pollutant:
            type: str
            required: True
            desc: "The air pollutant, e.g., fine particulate matter
              concentration"
            example: "PM2.5"
          months-back:
            type: int
            required: False
            desc: "Number of months back to include"
            example: "12"
      - map:
          name:
            type: str
            pattern: aggregate-data
            required: True
          aggregation-level:
            type: str
            required: True
            example: "city"
          aggregation-function:
            type: str
            required: False
            example: "mean"
      - map:
          name:

```



```

    type: str
    pattern: nop
    required: True
    desc: "Empty placeholder function not doing anything"
analysis:
  desc: "Data Analysis"
  required: True
  seq:
  - map:
    name:
      type: str
      pattern: trend-analysis
      required: True
    time-unit:
      type: str
      required: True
      example: "month"
    smoothing:
      type: str
      required: False
      example: "moving-average"
  - map:
    name:
      type: str
      pattern: seasonal-decomposition
      required: True
    frequency:
      type: int
      required: True
      desc: "e.g., 12 for monthly seasonality"
      example: "12"
    decomposition-method:
      type: str
      required: False
      example: "additive"
  - map:
    name:
      type: str
      pattern: nop
      required: True
      desc: "Empty placeholder function not doing anything"
visualization:
  desc: "Data Visualization"
  required: True
  seq:
  - map:
    name:
      type: str
      pattern: line-chart
      required: True

```

```

x-axis:
  type: str
  required: True
  example: "time"
y-axis-label:
  type: str
  required: False
  example: "PM2.5 Concentration"
- map:
  name:
    type: str
    pattern: heatmap
    required: True
  color-scale:
    type: str
    required: True
    example: "Reds"
  annotation:
    type: bool
    required: False
    desc: "A boolean value, e.g., True for displaying values"
    example: "True"
- map:
  name:
    type: str
    pattern: nop
    required: True
    desc: "Empty placeholder function not doing anything"
required: True

```

6. Example system message

Listing 3 shows the system message used in the prototype implementation. The schema and object YAML strings are dynamically interpolated with `${jsYaml.dump(objectSchema.value, {indent: 1})}` and `${jsYaml.dump(objectState.value, {indent: 1})}`. The backticks in the string are escaped with preceding backslashes, but the LLM does not see these backslashes. Note that the listing wraps text at linebreaks not present in the original text.

Listing 3: Example system message text. The full system message, including object and schema, has 1,557 tokens using GPT-3 tokenizer.

```

You are the assistant for the SchemaPrompt application. SchemaPrompt is a
  service that lets the users pick from a library of predefined
  functions and compose them into an analysis pipeline job that will
  later be submitted to a backend for scheduling and processing on a
  cluster, possibly returning a downloadable file archive.

```

You will assist the user by stepwise building the YAML structure that tells SchemaPrompt which functions to include in the analysis job.

Make sure to only use a YAML structure that is valid for the Kwalify schema. Never include elements to the YAML that violate the Kwalify schema's additional conditional logic or constraints.

This is the Kwalify schema:

```
\`\`\`  
${jsYaml.dump(objectSchema.value, {indent: 1})}\`\`\`
```

This initial YAML structure is available:

```
\`\`\`  
${jsYaml.dump(objectState.value, {indent: 1})}\`\`\`
```

You must output the complete current YAML structure this far for each response.

Further, never list the schema in response to the user. Instead of having the complete Kwalify schema in response to the user, provide the user with a human-readable description. Finally, do avoid the word schema in your answers.

You will only assist the user with building a correct and valid YAML structure and providing supporting information related to this process. You cannot assist or help the user with unrelated requests. When explaining to the user or making suggestions, only include valid options.

Only show the current YAML once per answer from the assistant. Always enclose the YAML string in `\`\`\``. Never show other YAML examples, and never add elements to the YAML unless the user has requested it. If in doubt, ask the user instead of assuming what to add. Don't add the schema's title, description, or examples meta-data fields to the YAML.

Instead of "YAML structure" or "YAML", you shall just use "specification".

Please answer this system message by welcoming the user and explaining the steps.

6.1. Extracting object updates

A fundamental element of our approach is that the LLM completions must contain an updated version of the object structure after modifications. The client application will then use a regular expression [23] to extract the text

representation of the object structure. The extracted text for the object can then be validated according to the schema. If the validation is successful, the client can update the current object it manages, possibly deserializing first. In our prototype, the client application instructs the LLM to always enclose the text for the object structure with triple back-ticks. For example, GPT-3.5 Turbo and GPT-4 usually do this by default because the web interface for ChatGPT supports Markdown [24] and a common extension to the syntax where triple back-ticks are used for fenced code blocks (<https://www.markdownguide.org/cheat-sheet/#extended-syntax>). Unfortunately, there are cases where GPT omits the backticks or uses backticks for other parts of the text completion. Thus, we must consider this when designing the regular expression used to match and extract the object structure text, which may require a portion of trial and error.

7. Experiment and results

This section reports the design and result of an experiment using a set of six prompts for assessing three different LLMs, GPT-4, GPT-3.5 Turbo, and GPT4All-J. The first two were accessed online programmatically via the OpenAI API, whereas GPT4All-J ran on a PC utilizing the graphical processing unit (GPU). The PC had an AMD Ryzen 9 7900X CPU, 64 GB RAM main memory, and an NVIDIA GeForce RTX 4080 graphics card with 16 GB memory.

For all the models, the temperature parameter value was set to 0.05, yielding more stable and less creative outputs. For GPT4All-J, we also set `max_length` to 2,000, `min_new_tokens` to 20, and `repetition_penalty` to 2.0. Details on the parameters are available at <https://platform.openai.com/docs/api-reference/completions/create> for the OpenAI API and at https://huggingface.co/docs/transformers/main_classes/text_generation#transformers.GenerationConfig for GPT4All-J.

7.1. Experiment design

To assess if the LLMs could fulfill the essential needs of our approach, we designed six prompts, each assessing different requirements. Further, in addition to responding correctly to the user, the YAML representing an updated object returned by the LLM was required to be correct. Table 1 lists what the six prompts assess, how it is assessed, and the success criteria. The actual prompt texts can be found in Table 2, and an example of an unsuccessful completion can be seen in Table 3.

Table 1: The table shows what the experiment’s six prompts assess and how.

| # | What to assess | How to assess | Success criterion |
|---|----------------|---------------|-------------------|
|---|----------------|---------------|-------------------|

| | | | |
|---|--|--|--|
| 1 | Does the LLM comprehend the overall task? | The last line of the initial system requests a welcome message (Listing 3). | The LLM provides an appropriate greeting and summary. |
| 2 | Has the LLM understood the schema rules? | The user asks for a user-friendly explanation of the schema. | The LLM provides an accurate, detailed description of the requirements of the Kwalify schema in natural, user-friendly language. |
| 3 | Does the LLM enforce the schema rules, such as mandatory parameters? | The user requests the assistant to add a function to the pipeline but does not provide the value for a required parameter. | The LLM should ask the user for the missing parameter value and not add the function until given. |
| 4 | Can the LLM interpret a prompt in the context of the previous prompt and completion, i.e., can the configuration process span over more than one step? | The user provides the missing parameter value. | The LLM correctly adds the function to the pipeline, including the required parameter value. |
| 5 | Does the LLM understand the schema rule dictating that some elements only can be added to a specific parent element? | The user asks to add an existing function to a step for which it is not valid. | The LLM should not add the function. |
| 6 | Does the LLM refrain from engaging in topics irrelevant to its task? | The user asks an off-topic question. | The LLM does not provide an answer to the question. |

7.2. Results

Table 2 shows the user prompts and the visible part of the completions returned by GPT-4, where the client application has removed any YAML representations of objects. Table 4 summarizes the results for the different models, which shows if the completions for GPT-4, GPT-3.5 Turbo, and GPT4All-J were acceptable given the prompt’s purpose.

Table 2: The table shows experiments with prompts and completion using GPT-4 as the system’s LLM

| 1: Welcome message |
|---|
| <p>System: Please answer this system message by welcoming the user and explaining the steps.</p> <p>Assistant: Welcome to SchemaPrompt! I’m here to help you build a specification for your analysis pipeline job. We will go through the following steps:</p> <p>1. Data Wrangling:</p> <ul style="list-style-type: none">• filter-data: Filters the dataset based on a specific pollutant and an optional number of months back.• pollutant: The air pollutant to filter by (e.g., PM2.5).• months-back: (Optional) Number of months back to include in the filtered data.• aggregate-data: Aggregates the dataset based on a specified level and an optional aggregation function.• aggregation-level: The level to aggregate the data by (e.g., city).• aggregation-function: (Optional) The function to use for aggregation (e.g., mean).• nop: An empty placeholder function that does nothing. |

1. Data Analysis:

- **trend-analysis:** Analyzes trends in the dataset based on a specified time unit and an optional smoothing method.
 - **time-unit:** The time unit for trend analysis (e.g., month).
 - **smoothing:** (Optional) The smoothing method to apply (e.g., moving-average).
- **seasonal-decomposition:** Decomposes the dataset into seasonal components based on a specified frequency and an optional decomposition method.
 - **frequency:** The frequency for seasonal decomposition (e.g., 12 for monthly seasonality).
 - **decomposition-method:** (Optional) The method for seasonal decomposition (e.g., additive).
- **nop:** An empty placeholder function that does nothing.

1. Data Visualization:

- **line-chart:** Creates a line chart with specified x-axis and an optional y-axis label.
 - **x-axis:** The variable for the x-axis (e.g., time).
 - **y-axis-label:** (Optional) The label for the y-axis (e.g., PM2.5 Concentration).
- **heatmap:** Generates a heatmap with a specified color scale and an optional annotation.
 - **color-scale:** The color scale for the heatmap (e.g., Reds).
 - **annotation:** (Optional) A boolean value indicating whether to display values on the heatmap (e.g., True).
- **nop:** An empty placeholder function that does nothing.

Please let me know which functions you'd like to include in each section, and I'll help you build the specification.

3: The request is missing a required parameter

User: filter data

| |
|--|
| <p>Assistant: To filter data, we need to specify the pollutant you want to filter. Please provide the pollutant (e.g., PM2.5) and optionally, the number of months back you want to include in the analysis. Once you provide this information, I will update the specification for you.</p> |
| <p>4: A continuation of the previous request</p> |
| <p>User: PM2.5 last 12 months</p> <p>Assistant: Great! I have added the filter-data function with the specified pollutant (PM2.5) and months-back (12) to the specification.</p> <p>Next, we can move on to data aggregation or proceed to the analysis and visualization steps. Let me know how you'd like to proceed.</p> |
| <p>5: Request to add a function to the wrong step</p> |
| <p>User: Add heatmap to analysis</p> <p>Assistant: Heatmap is a visualization function, not an analysis function. I can add it to the visualization section for you. Please provide the color scale you'd like to use for the heatmap (e.g., Reds) and optionally, if you want to display values as annotations (True or False). Once you provide this information, I will update the specification for you.</p> |
| <p>6: An off-topic request is made by the user</p> |
| <p>User: What is the population of the EU?</p> <p>assistant : I'm sorry, but I can only assist you with building a specification for your analysis pipeline job using SchemaPrompt. If you have any questions or need help with the specification, please let me know, and I'll be happy to help.</p> |

For GPT4All-J, we slightly modified the initial system message due to the API not providing a data structure for separating prompts and completions. We tried different ways of sectioning the message to exemplify the prompt and completion structure to the LLM, including separators and instructions to the LLM. Further, we experimented with parameters such as `temperature`, `repeat_penalty`, `top_k`, and `top_p`. Setting the repeat penalty higher than one was important to avoid GPT4All-J repeating a sentence in a loop. Unfortunately, the completions did not satisfy our requirements. The LLM also used around 54 to 56 seconds to respond with a completion. The answer closest to fulfilling the first requirements of the welcome message is shown in Table 3.

Table 3: The table shows the welcome message produced by GPT4All-J.

| 1: Welcome message |
|---|
| Assistant: Welcome User! Please follow these steps carefully while configuring Pipeline : 1) Selecting Data Source 2) Choosing Functions 3a.) Adding Filters 4b.). Aggregation 5c).Visualizations 6d)). Analyser 7finally 8). The final result should look like below image (with no errors): https://i1s4z0qwjnx9lk7v8h6m3pfg@bigasscluster/images?max_id=1568894567&autoplay=-1038993600 " |

Table 4: The table shows whether or not the completions for prompts 1–6 were acceptable (Y or N) for GPT-4, GPT-3.5 Turbo, and GPT4All-J.

| Prompt # | GPT-4 | GPT-3.5 Turbo | GPT4All-J |
|----------|-------|---------------|-----------|
| 1 | Y | Y | N |
| 2 | Y | Y | N |
| 3 | Y | Y | N |
| 4 | Y | Y | N |
| 5 | Y | Y | N |
| 6 | Y | Y | N |

8. Discussion

Because testing all responses to all possible ways for a user to prompt an LLM is infeasible, we accept that our experiment prompts cannot prove that our priming always works. Instead, we used minimal prompts designed to demonstrate that an LLM, such as GPT-4, provided our approach’s fundamental requirements. However, another mechanism for ensuring the correctness of the updated objects returned by the LLM is to employ schema validation after extracting the YAML (or other representation) from the complete text. Validation is the primary purpose of schemas. In addition, experiments have shown that results can be improved by asking LLMs to reflect on and correct wrong answers, termed the Reflexion technique [25]. Thus, if validation fails, the LLM may be able to correct itself if asked. However, we have not conducted such experiments here.

GPT-4 and GPT-3.5 Turbo met all requirements assessed by the prompts, but GPT4All-J did not, which may seem discouraging regarding democratizing

artificial intelligence in terms of open and free-to-use alternatives having a comparable quality to the closed commercial services. However, there is currently much voluntary effort to develop improved open LLM models that run locally. Hopefully, open models that can run locally and compete with GPT-3.5 and GPT-4 will eventually emerge.

9. Conclusion

We have presented an approach utilizing schemas included in prompts, enabling LLMs to comprehend and enforce a complex set of rules defining data objects' valid structure and content. The approach shows promising results in experiments for two of the three LLMs assessed. It can be applied when creating natural-language user interfaces for configuring jobs for backend systems or performing other tasks. However, the proposed approach may have the potential for a broader range of use cases.

10. Declarations

10.1. Competing interests

The authors have no competing interests to declare relevant to this article's content.

References

- [1] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, B. Ge, Summary of chatgpt/gpt-4 research and perspective towards the future of large language models (4 2023). doi:10.48550/arxiv.2304.01852.
- [2] OpenAI, Gpt-4 technical report, Report, OpenAI (3 2023). doi:10.48550/arxiv.2303.08774.
- [3] OpenAI, [Introducing chatgpt](https://openai.com/blog/chatgpt), accessed: 2023-04-17 (11 2022). URL <https://openai.com/blog/chatgpt>
- [4] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, Adaptive computation and machine learning, MIT Press, Cambridge, Mass, 2016.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010.
- [6] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, [Improving language understanding by generative pre-training](https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf), accessed: 2023-04-20 (2018). URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

- [7] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback (3 2022). doi:10.48550/arXiv.2203.02155.
- [8] What are tokens and how to count them?, accessed: 2023-04-17 (2023). URL <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>
- [9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models (2 2023). doi:10.48550/arXiv.2302.13971.
- [10] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, E. P. Xing, Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, accessed: 2023-04-17 (3 2023). URL <https://vicuna.lmsys.org>
- [11] B. Wang, A. Komatsuzaki, GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model, accessed: 2023-04-17 (5 2021). URL <https://github.com/kingoflolz/mesh-transformer-jax>
- [12] M. Conover, M. Hayes, A. Mathur, X. Meng, J. Xie, J. Wan, A. Ghodsi, P. Wendell, M. Zaharia, Hello dolly: Democratizing the magic of chatgpt with open models, accessed: 2023-04-17 (3 2023). URL <https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html>
- [13] Y. Anand, Z. Nussbaum, B. Duderstadt, B. M. Schmidt, A. Treat, A. Mulyar, Gpt4all-j: An apache-2 licensed assistant-style chatbot, accessed: 2023-04-17 (2023). URL https://static.nomic.ai/gpt4all/2023_GPT4All-J_Technical_Report_2.pdf
- [14] M. Conover, M. Hayes, A. Mathur, X. Meng, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, R. Xin, Free dolly: Introducing the world’s first truly open instruction-tuned llm, accessed: 2023-04-17 (4 2023). URL <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>
- [15] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal, Pythia: A suite for analyzing large language models across training and scaling (4 2023). doi:10.48550/arXiv.2304.01373.

- [16] R. Albergotti, [The secret history of elon musk, sam altman, and openai](#), Semafor.
URL <https://www.semafor.com/article/03/24/2023/the-secret-history-of-elon-musk-sam-altman-and-openai>
- [17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models (6 2021).
[doi:10.48550/arXiv.2106.09685](https://doi.org/10.48550/arXiv.2106.09685).
- [18] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. El-nashar, J. Spencer-Smith, D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with chatgpt (2 2023). [doi:10.48550/arXiv.2302.11382](https://doi.org/10.48550/arXiv.2302.11382).
- [19] J. Edelman, S. S. Lowe, M. Oswalt, Network Programmability and Automation, O'Reilly, Sebastopol, CA, 2018.
- [20] A. Wařowski, Domain-specific languages: Effective modeling, automation, and reuse (2023).
- [21] OpenAI, [Openai plugins](#), accessed: 2023-04-17 (3 2023).
URL <https://openai.com/blog/chatgpt-plugins>
- [22] G. Brockman, P. Welinder, M. Murati, OpenAI, [Openai api](#), accessed: 2023-04-17 (6 2020).
URL <https://openai.com/blog/openai-api>
- [23] J. E. Friedl, Mastering regular expressions (2006).
- [24] J. Gruber, [Introducing markdown](#), accessed: 2023-04-17 (3 2004).
URL https://daringfireball.net/2004/03/introducing_markdown
- [25] N. Shinn, B. Labash, A. Gopinath, Reflexion: an autonomous agent with dynamic memory and self-reflection (3 2023). [doi:10.48550/arXiv.2303.11366](https://doi.org/10.48550/arXiv.2303.11366).

Bibliography

- Agerbo, Esben, Merete Nordentoft, and Preben B. Mortensen (2002). “Familial, psychiatric, and socioeconomic risk factors for suicide in young people: nested case-control study.” In: *BMJ* 325.7355, pp. 74–77.
- Aikens, Rachael C. et al. (2021). “stratamatch: Prognostic Score Stratification Using a Pilot Design.” In: *The R Journal* 13.1, pp. 614–630.
- Albergotti, Reed (Mar. 2023). “The Secret History of Elon Musk, Sam Altman, and OpenAI.” In: *Semafor*. URL: <https://www.semafor.com/article/03/24/2023/the-secret-history-of-elon-musk-sam-altman-and-openai>.
- Alexander, Lorraine K. et al. (2015a). *Confounding Bias, Part I*. UNC Gillings School of Global Public Health. URL: https://sph.unc.edu/wp-content/uploads/sites/112/2015/07/nciph_ERIC11.pdf (visited on 06/25/2023).
- (2015b). *Confounding Bias, Part II and Effect Measure Modification*. UNC Gillings School of Global Public Health. URL: https://sph.unc.edu/wp-content/uploads/sites/112/2015/07/nciph_ERIC12.pdf (visited on 06/25/2023).
 - (2015c). *Incident vs. Prevalent Cases and Measures of Occurrence*. UNC Gillings School of Global Public Health. URL: https://sph.unc.edu/wp-content/uploads/sites/112/2015/07/nciph_ERIC1.pdf (visited on 06/25/2023).
 - (2015d). *Randomized Controlled Trials (Experimental Studies)*. UNC Gillings School of Global Public Health. URL: https://sph.unc.edu/wp-content/uploads/sites/112/2015/07/nciph_ERIC10.pdf (visited on 06/19/2023).
- Ambler, Gareth, Axel Benner, and Georg Heinze (2022). *mfp: Multivariable Fractional Polynomials*. URL: <https://CRAN.R-project.org/package=mfp> (visited on 03/03/2023).

- Anand, Yuvanesh et al. (2023). *GPT4All-J: An Apache-2 Licensed Assistant-Style Chatbot*. URL: https://static.nomic.ai/gpt4all/2023_GPT4All-J_Technical_Report_2.pdf (visited on 04/17/2023).
- Aycock, John (2003). "A Brief History of Just-in-Time." In: *ACM Comput Surv* 35.2, pp. 97–113.
- Baglietto, Laura et al. (2017). "DNA Methylation Changes Measured in Pre-Diagnostic Peripheral Blood Samples Are Associated With Smoking and Lung Cancer Risk." In: *Int J Cancer* 140.1, pp. 50–61.
- Baiju, Nikita et al. (2021). "Gene Expression in Blood Reflects Smoking Exposure Among Cancer-Free Women in the Norwegian Women and Cancer (NOWAC) Postgenome Cohort." In: *Sci Rep* 11.
- Baker, Monya (2016). "1,500 Scientists Lift the Lid on Reproducibility." In: *Nature* 533.7604, pp. 452–454.
- Baldassari-Mather, Lois (1992). "Use of a Visually Oriented Software Development Tool for Creating Automated Test Programs." In: *[1992] Conference Record IEEE Instrumentation and Measurement Technology Conference*, pp. 249–254.
- Barham, Paul et al. (2003). "Xen and the Art of Virtualization." In: *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*. SOSP '03. ACM, pp. 164–177.
- Barnung, Runa B. et al. (2018). "Coffee Consumption and Whole-Blood Gene Expression in the Norwegian Women and Cancer Post-Genome Cohort." In: *Nutrients* 10.8.
- Beck, Micah (2019). "On The Hourglass Model." In: *Commun ACM* 62.7, pp. 48–57.
- Behjati, Sam and Patrick S. Tarpey (2013). "What Is Next Generation Sequencing?" In: 98.6, pp. 236–238.
- Biderman, Stella et al. (Apr. 2023). "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling." In: *arXiv*. DOI: 10.48550/arXiv.2304.01373.
- Bingham, Sheila and Elio Riboli (2004). "Diet and Cancer—The European Prospective Investigation Into Cancer and Nutrition." In: *Nat Rev Cancer* 4.3, pp. 206–215.

- Blackwell, Matthew et al. (2009). “CEM: Coarsened Exact Matching in Stata.” In: *The Stata Journal* 9.4, pp. 524–546.
- Brambilla, Marco, Jordi Cabot, and Manuel Wimmer (2012). *Model-Driven Software Engineering in Practice*. Morgan & Claypool.
- Breschi, Alessandra, Thomas R. Gingeras, and Roderic Guigó (2017). “Comparative Transcriptomics in Human and Mouse.” In: *Nat Rev Genet* 18.7, pp. 425–440.
- Brockman, Greg et al. (June 2020). *OpenAI API*. URL: <https://openai.com/blog/openai-api> (visited on 04/17/2023).
- Buis, Maarten L. (2009). “Using and Interpreting Restricted Cubic Splines.” In: Bonn, Germany: 7th German Stata Users Group Meeting. URL: <http://www.maartenbuis.nl/presentations/bonn09.pdf> (visited on 03/03/2023).
- Caetano-Anolles, Derek (2023a). *About the GATK Best Practices*. Broad Institute. URL: <https://gatk.broadinstitute.org/hc/en-us/articles/360035894711-About-the-GATK-Best-Practices> (visited on 07/13/2023).
- (2023b). *Pipelining GATK with WDL and Cromwell*. Broad Institute. URL: <https://gatk.broadinstitute.org/hc/en-us/articles/360035889771-Pipelining-GATK-with-WDL-and-Cromwell> (visited on 07/13/2023).
- Campbell, Neil A., Jane B. Reece, and Eric J. Simon (2007). *Essential Biology*. 3rd int. Pearson Benjamin Cummings.
- Castagne, Raphael et al. (2016). “Biological Marks of Early-Life Socioeconomic Experience Is Detected in the Adult Inflammatory Transcriptome.” In: *Sci Rep* 6.1, p. 38705.
- Chatfield, Chris (1995). “Model Uncertainty, Data Mining and Statistical Inference.” In: *J R Stat Soc Ser A Stat Soc* 158.3, pp. 419–466.
- Chiang, Wei-Lin et al. (Mar. 2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. URL: <https://vicuna.lmsys.org> (visited on 04/17/2023).
- Cockburn, Alistair (2000). *Writing Effective Use Cases*. Addison-Wesley.
- Coggon, David, David Barker, and Geoffrey Rose (2009). *Epidemiology for the Uninitiated*. 5th ed. Wiley.

- Conover, Mike, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, et al. (Mar. 2023). *Hello Dolly: Democratizing the magic of ChatGPT with open models*. URL: <https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html> (visited on 04/17/2023).
- Conover, Mike, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, et al. (Apr. 2023). *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM*. URL: <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm> (visited on 04/17/2023).
- Cooper, Diane (2016). "Evidence ladder and the Journal of the Medical Library Association." In: *J Med Libr Assoc* 104.4, p. 262.
- Cormen, Thomas H. et al. (2022). *Introduction to Algorithms*. 4th. MIT.
- Dahlquist, Germund and Åke Björck (1974). *Numerical Methods*. Prentice-Hall.
- Dammann, Olaf et al. (2014). "Systems Epidemiology: What's in a Name?" In: *Online J Public Health Inform* 6.3, e198.
- de Visser, Karin E., Alexandra Eichten, and Lisa M. Coussens (2006). "Paradoxical Roles of the Immune System During Cancer Development." In: *Nat Rev Cancer* 6.1, pp. 24–37.
- Du, Zhanwei et al. (2023). "Editorial: Infectious Disease Epidemiology and Transmission Dynamics." In: *Viruses* 15.1, p. 246.
- Dumeaux, Vanessa, Anne-Lise Børresen-Dale, et al. (2008). "Gene Expression Analyses in Breast Cancer Epidemiology: The Norwegian Women and Cancer Postgenome Cohort Study." In: *Breast Cancer Res* 10.1, R13.
- Dumeaux, Vanessa, Bjørn Fjukstad, et al. (2017). "Interactions Between the Tumor and the Blood Systemic Response of Breast Cancer Patients." In: *PLoS Comput Biol* 13.9, e1005680.
- Edelman, Jason, Scott S. Lowe, and Matt Oswalt (2018). *Network Programmability and Automation*. O'Reilly.
- Eilers, Paul H. C. and Brian D. Marx (2021). *Practical Smoothing: The Joys of P-splines*. Cambridge University.

- Eilers, Paul H. C., Brian D. Marx, and Maria Durbán (2015). “Twenty years of P-splines.” In: *SORT* 39.2, pp. 149–186.
- Fasanelli, Francesca et al. (2015). “Hypomethylation of Smoking-Related Genes Is Associated With Future Lung Cancer in Four Prospective Cohorts.” In: *Nat Commun* 6.1, p. 10192.
- Federal Trade Commission Enforcement of the U.S.-EU and U.S.-Swiss Safe Harbor Frameworks* (2016). Federal Trade Commission. URL: <https://www.ftc.gov/business-guidance/resources/federal-trade-commission-enforcement-us-eu-us-swiss-safe-harbor-frameworks> (visited on 07/13/2023).
- Fink, Gil and Ido Flatow (2014). “Introducing Single Page Applications.” In: *Pro Single Page Application Development: Using Backbone.js and ASP.NET*. Apress, pp. 3–13.
- Fjukstad, Bjørn (2019). *Toward Reproducible Analysis and Exploration of High-Throughput Biological Datasets*. UiT The Arctic University of Norway.
- Fjukstad, Bjørn et al. (2019). “Reproducible Data Management and Analysis Using R.” In: *bioRxiv*. DOI: 10.1101/644625. URL: <http://biorxiv.org/content/early/2019/05/21/644625.abstract>.
- Foster, Ian (1995). *Designing and Building Parallel Programs*. Addison-Wesley.
- Foulds, L. (1958). “The Natural History of Cancer.” In: *J Chronic Dis* 8.1, pp. 2–37.
- Fowler, Martin (2005). *FluentInterface*. URL: <https://martinfowler.com/bliki/FluentInterface.html> (visited on 06/06/2019).
- Freedman, David and Persi Diaconis (1981). “On the Histogram as a Density Estimator: L2 theory.” In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57.4, pp. 453–476.
- Frieden, Thomas R. (2017). “Evidence for Health Decision Making — Beyond Randomized, Controlled Trials: The Changing Face of Clinical Trials.” In: *NEJM* 377.5, pp. 465–475.
- Friedl, Jeffrey E. F. (2006). *Mastering Regular Expressions*. 3rd ed. O’Reilly.
- Frost, Amy (2022). *Constitutional (Germline) vs Somatic (Tumour) Variants*. NHS England. URL: <https://www.genomicseducation.hee.nhs.uk/genotes/>

knowledge-hub/constitutional-germline-vs-somatic-tumour-variants/
(visited on 06/26/2023).

Fryling, Meg (2019). “Low Code App Development.” In: *J Comput Sci Coll* 34.6, p. 119.

García-Campos, Miguel A., Jesús Espinal-Enríquez, and Enrique Hernández-Lemus (2015). “Pathway Analysis: State of the Art.” In: *Front Physiol* 6, p. 383.

Gauthier, J., Q. V. Wu, and T. A. Gooley (2020). “Cubic Splines to Model Relationships Between Continuous Variables and Outcomes: A Guide for Clinicians.” In: *Bone Marrow Transplant* 55.4, pp. 675–680.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. Cambridge, Mass: MIT.

Gordis, Leon (2014). *Epidemiology*. Elsevier.

Gram, Inger T. et al. (2013). “The Hazards of Death by Smoking in Middle-Aged Women.” In: *Eur J Epidemiol* 28.10, pp. 799–806.

Grizzi, Fabio and Maurizio Chiriva-Internati (2006). “Cancer: Looking for Simplicity and Finding Complexity.” In: *Cancer Cell Int* 6.1, p. 4.

Gruber, John (Mar. 2004). *Introducing Markdown*. URL: https://daringfireball.net/2004/03/introducing_markdown (visited on 04/17/2023).

Hariton, Eduardo and Joseph J. Locascio (2018). “Randomised Controlled Trials: The Gold Standard for Effectiveness Research.” In: *BJOG* 125.13, pp. 1716–1716.

Harrell Jr., Frank E. (2015). *Regression Modeling Strategies*. 2nd. Springer.

Hasin, Yehudit, Marcus Seldin, and Aldons Lusic (2017). “Multi-Omics Approaches to Disease.” In: *Genome Biol* 18.1, p. 83.

Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman (2009). *The Elements of Statistical Learning*. 2nd. Springer.

Helseregisterloven (2022). The Norwegian Government. URL: <https://www.regjeringen.no/no/dokumenter/helseregisterloven/id2904352/> (visited on 07/13/2023).

- Hill, Austin B. (1965). “The Environment and Disease: Association or Causation?” In: *Proc R Soc Med* 58.5, pp. 295–300.
- Holden, Lars (2015). *Classify Strata*. Report SAMBA/11/15. Norwegian Computing Center. URL: <https://www.nr.no/files/samba/smbi/note2015SAMBA1115classifyStrata.pdf>.
- Hollingsworth, David (1995). *The Workflow Reference Model*. Workflow Management Coalition.
- Holsbø, Einar J. (2019). *Small Data: Practical Modeling Issues in Human-Model-Omic Data*. UiT The Arctic University of Norway.
- Hothorn, Torsten et al. (2005). “The Design and Analysis of Benchmark Experiments.” In: *J Comput Graph Stat* 14.3, pp. 675–699.
- Howards, Penelope P. (2018). “An Overview of Confounding. Part 1: The Concept and How to Address It.” In: *Acta Obstet Gynecol Scand* 97.4, pp. 394–399.
- Hu, Edward J. et al. (June 2021). “LoRA: Low-Rank Adaptation of Large Language Models.” In: *arXiv*. DOI: 10.48550/arXiv.2106.09685.
- Iacus, Stefano M., Gary King, and Giuseppe Porro (2008). “Matching for Causal Inference Without Balance Checking.” In: *SSRN*. URL: <https://ssrn.com/abstract=1152391>.
- (2009). “cem: Software for Coarsened Exact Matching.” In: *J Stat Softw* 30.9, pp. 1–27.
- IBM Cloud Team (2021). *SOA vs. Microservices: What’s the Difference?* IBM Cloud. URL: <https://www.ibm.com/cloud/blog/soa-vs-microservices> (visited on 06/29/2023).
- Id-Lung* (2023). UiT The Arctic University of Norway. URL: https://en.uit.no/forskning/forskningsgrupper/gruppe?p_document_id=507532 (visited on 07/07/2023).
- Institute of Systems and Synthetic Biology (2019). *About the Institute*. Imperial College London. URL: <https://www.imperial.ac.uk/systems-biology/about-the-institute/> (visited on 06/06/2019).
- Jackson, Brad et al. (2005). “An Algorithm for Optimal Partitioning of Data on an Interval.” In: *IEEE Signal Process Lett* 12.2, pp. 105–108.

- Joffe, Michael et al. (2012). “Causal Diagrams in Systems Epidemiology.” In: *Emerg Themes Epidemiol* 9.1, p. 1.
- Kim, Soyoung (2016). “Case-Cohort Studies vs Nested Case-Control Studies.” In: *Datum Newsletter Division of Biostatistics* 22.1, pp. 1–2.
- King, Gary and Richard Nielsen (2019). “Why Propensity Scores Should Not Be Used for Matching.” In: *Polit Anal* 27.4, pp. 435–454.
- Kleinbaum, David G. (2014). *Applied Regression Analysis and Other Multivariable Methods*. 5th. Cengage.
- Kleinbaum, David G., Kevin M. Sullivan, and Nancy D. Barker (2007). *Pocket Guide to Epidemiology*. Springer.
- Kleppman, Martin (2017). *Designing Data-Intensive Applications*. 1st. O’Reilly.
- Koch, Dirk, Daniel Ziener, and Frank Hannig (2016). *FPGAs for Software Programmers*. Springer.
- Kvinner og kreft: Blodprøve og biopsi* (2020). The Norwegian Women and Cancer Study. URL: <https://web.archive.org/web/20220626074511/https://site.uit.no/kvinnerogkreft/blodprove-og-biopsi/> (visited on 07/07/2023).
- Lanzing, Jan (1998). “Concept Mapping: Tools for Echoing the Minds Eye.” In: *J Vis Lit* 18.1, pp. 1–14.
- Larman, Craig (2002). *Applying UML and Patterns*. Prentice Hall.
- Leffondré, Karen et al. (2006). “Modelling smoking history using a comprehensive smoking index: application to lung cancer.” In: *Stat Med* 25.24, pp. 4132–4146.
- Lieb, Roselind (2013). “Population-Based Study.” In: *Encyclopedia of Behavioral Medicine*. Ed. by Marc D. Gellman and J. Rick Turner. Springer, pp. 1507–1508.
- Liu, Yiheng et al. (Apr. 2023). “Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models.” In: *arXiv*. DOI: 10.48550/arxiv.2304.01852.
- Lund, Eiliv and Vanessa Dumeaux (2008). “Systems Epidemiology in Cancer.” In: *Cancer Epidemiol Biomarkers Prev* 17.11, pp. 2954–7.

- Lund, Eiliv, Vanessa Dumeaux, et al. (2008). “Cohort Profile: The Norwegian Women and Cancer Study–NOWAC–Kvinner OG Kreft.” In: *Int J Epidemiol* 37.1, pp. 36–41.
- Lund, Eiliv, Lars Holden, et al. (2016). “A New Statistical Method for Curve Group Analysis of Longitudinal Gene Expression Data Illustrated for Breast Cancer in the NOWAC Postgenome Cohort as a Proof of Principle.” In: *BMC Med Res Methodol* 16.1, p. 28.
- Lund, Eiliv, Sandra Plancade, et al. (2015). “A Processual Model for Functional Analyses of Carcinogenesis in the Prospective Cohort Design.” In: *Med Hypotheses* 85.4, pp. 494–7.
- Luo, Zhen and Grace Wahba (1997). “Hybrid Adaptive Splines.” In: *J Am Stat Assoc* 92.437, pp. 107–116.
- Marinescu, Dan C. (2023). *Cloud Computing*. Morgan Kaufmann.
- Massie, Matt et al. (2013). *ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing*. Berkeley EECS.
- Mell, Peter M. and Timothy Grance (2011). *SP 800-145. The NIST Definition of Cloud Computing*. Tech. rep.
- Mestas, Javier and Christopher C. W. Hughes (2004). “Of Mice and Not Men: Differences Between Mouse and Human Immunology.” In: *J Immunol* 172.5, pp. 2731–2738.
- Miller, Ed (2022a). *DNA*. NHS England. URL: <https://www.genomicseducation.hee.nhs.uk/genotes/knowledge-hub/dna/> (visited on 06/26/2023).
- (2022b). *Gene*. NHS England. URL: <https://www.genomicseducation.hee.nhs.uk/genotes/knowledge-hub/gene/> (visited on 06/26/2023).
- Miloslavskaya, Natalia and Alexander Tolstoy (2016). “Big Data, Fast Data and Data Lake Concepts.” In: *Procedia Comput Sci* 88, pp. 300–305.
- Nadeem, Anas and Muhammad Zubair Malik (2022). “Case for Microservices Orchestration Using Workflow Engines.” In: *2022 IEEE/ACM 44th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pp. 6–10.

- National Human Genome Research Institute (2018). *Human Genome Project Timeline*. NIH. URL: <https://www.genome.gov/human-genome-project/Timeline-of-Events> (visited on 06/12/2019).
- (2019). *DNA Sequencing Costs: Data*. NIH. URL: <https://www.genome.gov/sequencingcostsdata> (visited on 06/11/2019).
- Ngo, Long H. et al. (2017). “Methodologic Considerations in the Design and Analysis of Nested Case-Control Studies: Association Between Cytokines and Postoperative Delirium.” In: *BMC Med Res Methodol* 17.1, p. 88.
- Olsen, Karina Standahl et al. (2021). “Global Blood Gene Expression Profiles Following a Breast Cancer Diagnosis-Clinical Follow-Up in the NOWAC Post-Genome Cohort.” In: *PloS One* 16.3, e0246650–e0246650.
- OpenAI (Nov. 2022). *Introducing ChatGPT*. URL: <https://openai.com/blog/chatgpt> (visited on 04/17/2023).
- (Mar. 2023a). *GPT-4 Technical Report*. Report. OpenAI. DOI: 10.48550/arxiv.2303.08774.
- (Mar. 2023b). *OpenAI Plugins*. URL: <https://openai.com/blog/chatgpt-plugins> (visited on 04/17/2023).
- Ouyang, Long et al. (Mar. 2022). “Training Language Models To Follow Instructions With Human Feedback.” In: *arXiv*. DOI: 10.48550/arXiv.2203.02155.
- Pacheco, Peter S. (2011). *An Introduction to Parallel Programming*. 5th ed. Burlington, MA: Morgan Kaufmann.
- Pasientjournalloven* (2022). The Norwegian Government. URL: <https://www.regjeringen.no/no/dokumenter/pasientjournalloven/id2904354/> (visited on 07/13/2023).
- Perperoglou, Aris et al. (2019). “A Review of Spline Function Procedures in R.” In: *BMC Med Res Methodol* 19.1, p. 46.
- Pinho, Daniel, Ademar Aguiar, and Vasco Amaral (2023). “What About the Usability in Low-Code Platforms? A Systematic Literature Review.” In: *J Comput Lang* 74.
- Porta, Miquel (2014). *A Dictionary of Epidemiology*. 6th ed. Oxford.

- Pourmirza, Shaya et al. (2017). “A Systematic Literature Review on the Architecture of Business Process Management Systems.” In: 66, pp. 43–58.
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. URL: <https://www.R-project.org/> (visited on 03/03/2023).
- Radford, Alec, Jong Wook Kim, et al. (23–29 Jul 2023). “Robust Speech Recognition via Large-Scale Weak Supervision.” In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 28492–28518.
- Radford, Alec, Karthik Narasimhan, et al. (2018). *Improving Language Understanding by Generative Pre-Training*. OpenAI. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (visited on 04/20/2023).
- Ranganathan, Priya and Rakesh Aggarwal (2019). “Study Designs: Part 3 - Analytical Observational Studies.” In: *Perspect Clin Res* 10.2, pp. 91–94.
- Regulations* (2018). The Norwegian Data Protection Authority. URL: <https://www.datatilsynet.no/en/regulations-and-tools/regulations/> (visited on 07/13/2023).
- Richardson, Clay et al. (2014). *New Development Platforms Emerge For Customer-Facing Applications*. Forrester.
- Richter, Felix (2023). *Big Three Dominate the Global Cloud Market*. Statista. URL: <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/> (visited on 07/13/2023).
- Royston, Patrick and Douglas G. Altman (1994). “Regression Using Fractional Polynomials of Continuous Covariates: Parsimonious Parametric Modelling.” In: *J R Stat Soc Ser C Appl Stat* 43.3, pp. 429–467.
- Runkler, Thomas A. (2020). *Data Analytics: Models and Algorithms for Intelligent Data Analysis*. Springer.
- Russell, Stuart and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. 3rd int. Pearson.
- Sackett, David L. (1979). “Bias in Analytic Research.” In: *J Chronic Dis* 32.1, pp. 51–63.

- Sadalage, Pramod J. and Martin Fowler (2013). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Pearson.
- Sauerbrei, Willi et al. (2014). “STREngthening Analytical Thinking for Observational Studies: The STRATOS Initiative.” In: *Stat Med* 33.30, pp. 5413–5432.
- Scargle, Jeffrey D. et al. (2013). “Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations.” In: *Astrophys J* 764.2, p. 167.
- Schwarz, Gideon (1978). “Estimating the Dimension of a Model.” In: *Ann Stat* 6.2, pp. 461–464.
- Scott, David W. (1979). “On Optimal and Data-Based Histograms.” In: *Biometrika* 66.3, pp. 605–610.
- Sedgewick, Robert (1998). *Algorithms in C: Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*. 3rd. Addison-Wesley.
- Sekhon, Jasjeet S. (2008). “The Neyman-Rubin Model of Causal Inference and Estimation via Matching Methods.” In: ed. by Janet M. Box-Steffensmeier, Henry E. Brady, and David Collier. Oxford University, pp. 271–299.
- Shimazaki, Hideaki and Shigeru Shinomoto (2007). “A Method for Selecting the Bin Size of a Time Histogram.” In: *Neural Comput* 19.6, pp. 1503–1527.
- Shinn, Noah, Beck Labash, and Ashwin Gopinath (Mar. 2023). “Reflexion: an autonomous agent with dynamic memory and self-reflection.” In: *arXiv*. DOI: 10.48550/arXiv.2303.11366.
- Steyerberg, Ewout W. (2019). *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. 2nd. Springer.
- Stone, Charles J. (1986). “Comment: Generalized Additive Models.” In: *Stat Sci* 1.3, pp. 312–314.
- Straif-Bourgeois, Susanne, Raoult Ratard, and Mirjam Kretzschmar (2014). “Infectious Disease Epidemiology.” In: *Handbook of Epidemiology*. Ed. by Wolfgang Ahrens and Iris Pigeot. Springer, pp. 2041–2119.
- Strasser, Thomas (1985). “Implications of Epidemiology for Community Medicine.” In: *Prev Med* 12.1, pp. 150–154.

- Sturges, Herbert A. (1926). "The Choice of a Class Interval." In: *J Am Stat Assoc* 21.153, pp. 65–66.
- Sugerman, Jeremy, Ganesh Venkitachalam, and Beng-Hong Lim (2001). "Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor." In: *2001 USENIX Annual Technical Conference (USENIX ATC 01)*. USENIX.
- Syriani, Eugene, Lechanceux Luhunu, and Houari Sahraoui (2018). "Systematic Mapping Study of Template-Based Code Generation." In: *Comput Lang Syst | Struct* 52, pp. 43–62.
- Tanenbaum, Andrew S. and Maarten Van Steen (2014). *Distributed Systems: Principles and Paradigms*. 2nd ed. Pearson.
- The Health Research Act* (2020). National Research Ethics Committees. URL: <https://www.forskningsetikk.no/en/resources/the-research-ethics-library/legal-statutes-and-guidelines/the-health-research-act/> (visited on 07/13/2023).
- Timeline* (2018). The Norwegian Women and Cancer Study. URL: <https://web.archive.org/web/20220626031954/https://site.uit.no/nowac/methodological-description/timeline/> (visited on 05/30/2023).
- Torgersen, Mads (2007). "Querying in C#: How Language Integrated Query (LINQ) Works." In: *Companion to the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications Companion*. OOPSLA '07. ACM, pp. 852–853.
- Touvron, Hugo et al. (Feb. 2023). "LLaMA: Open and Efficient Foundation Language Models." In: *arXiv*. DOI: 10.48550/arXiv.2302.13971.
- Tripepi, Giovanni et al. (2010). "Stratification for Confounding - Part 1: The Mantel-Haenszel Formula." In: *Nephron Clin Pract* 116.4, pp. 317–321.
- Update on the Privacy Shield Framework* (2020). Federal Trade Commission. URL: <https://www.ftc.gov/business-guidance/privacy-security/privacy-shield> (visited on 07/13/2023).
- Vailati-Riboni, Mario, Valentino Palombo, and Juan J. Loor (2017). "What Are Omics Sciences?" In: *Periparturient Diseases of Dairy Cows: A Systems Biology Approach*. Ed. by Burim N. Ametaj. Springer, pp. 1–7.

- van der Wel, Kjetil A. et al. (2019). “A Gold Mine, but Still No Klondike: Nordic Register Data in Health Inequalities Research.” In: *Scand J Public Health* 47.6, pp. 618–630.
- van Veldhoven, Karin et al. (2015). “Epigenome-Wide Association Study Reveals Decreased Average Methylation Levels Years Before Breast Cancer Diagnosis.” In: *Clin Epigenetics* 7.1, p. 67.
- Variabler* (2023). Helsedata. URL: https://helsedata.no/no/variabler/?datakilde=K_KK&page=search (visited on 05/30/2023).
- Vaswani, Ashish et al. (2017). “Attention is All You Need.” In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates, pp. 6000–6010.
- Wacholder, Sholom et al. (1992a). “Selection of Controls in Case-Control Studies — I. Principles.” In: *Am J Epidemiol* 135.9, pp. 1019–1028.
- (1992b). “Selection of Controls in Case-Control Studies — II. Types of Controls.” In: *Am J Epidemiol* 135.9, pp. 1029–1041.
- (1992c). “Selection of Controls in Case-Control Studies — III. Design Options.” In: *Am J Epidemiol* 135.9, pp. 1042–1050.
- Wang, Ben and Aran Komatsuzaki (May 2021). *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. URL: <https://github.com/kingoflolz/mesh-transformer-jax> (visited on 04/17/2023).
- Wąsowski, Andrzej (2023). *Domain-Specific Languages: Effective Modeling, Automation, and Reuse*. Springer.
- Watt, D. C. et al. (1996). “Survival Analysis: The Importance of Censored Observations.” In: *Melanoma Res* 6.5, pp. 379–85.
- What Are Tokens and How To Count Them?* (2023). URL: <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them> (visited on 04/17/2023).
- White, Jules et al. (Feb. 2023). “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT.” In: *arXiv*. DOI: 10.48550/arXiv.2302.11382.
- Wilkinson, Barry and Michael Allen (2005). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. 2nd ed. Pearson/Prentice Hall.

- Xie, Yihui, J. J. Allaire, and Garrett Golemund (2019). *R Markdown: The Definitive Guide*. 1st. Routledge, pp. 63–78.
- Zaharia, Matei, Mosharaf Chowdhury, Tathagata Das, et al. (2012). “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing.” In: *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, p. 2.
- Zaharia, Matei, Mosharaf Chowdhury, Michael J. Franklin, et al. (2010). “Spark: Cluster Computing with Working Sets.” In: *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*. HotCloud’10. USENIX Association, p. 10.
- Zucchini, Walter (2000). “An Introduction to Model Selection.” In: *J Math Psychol* 44.1, pp. 41–61.

